

Technical Report 362

# Qualitative Knowledge, Causal Reasoning, and the Localization of Failures

Allen Brown

MIT Artificial Intelligence Laboratory

*This blank page was inserted to preserve pagination.*

**Qualitative Knowledge, Causal Reasoning,  
and the Localization of Failures**

by

Allen Leon Brown, Jr.  
Massachusetts Institute of Technology  
November 1976

Revised version of a dissertation submitted to the Department of Electrical Engineering and Computer Science on September 3, 1975 in partial fulfillment of the requirements of the Degree of Doctor of Philosophy.

*This empty page was substituted for a  
blank page in the original document.*

## **Abstract**

This report investigates some techniques appropriate to representing the knowledge necessary for understanding a class of electronic machines -- radio receivers. A computational performance model - WATSON - is presented. WATSON's task is to isolate failures in radio receivers whose principles of operation have been appropriately described in his knowledge base. The thesis of the report is that hierarchically organized representational structures are essential to the understanding of complex mechanisms. Such structures lead not only to descriptions of machine operation at many levels of detail, but also offer a powerful means of organizing "specialist" knowledge for the repair of machines when they are broken.

**Thesis Supervisor:** Gerald J. Sussman

**Title:** Assistant Professor of Electrical Engineering and Computer Science

### **Acknowledgments**

Many thanks are due to Collette, Martin, Mary, Don McDermott, and Gerald Susman for their contributions to the technical content of this report.

O spensierare della pace,  
non si fidare di nessuno in pace che non sia armato  
per la difesa della pace.  
Ma non si fidare di nessuno in pace  
che non sia armato con la pace.

— Leonardo da Vinci

*This empty page was substituted for a  
blank page in the original document.*



## Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Why radios? .....	1
1.2	A theory of machine description. ....	6
1.3	Where do plans and designs come from? .....	8
1.4	A theory of machine diagnosis. ....	10
1.5	Related work. ....	13
1.6	What is to come. ....	17
 <b>2</b>	 <b>Scenarios of WATSON's Performance</b>	 <b>18</b>
2.1	A opened junction in a transistor. ....	20
2.2	A shorted capacitor. ....	29
2.3	A misaligned front end. ....	33
2.4	An opened collector-base junction. ....	36
 <b>3</b>	 <b>A Preview of the WATSON Program</b>	 <b>42</b>
3.1	Stating the problem. ....	42
3.2	Localizing the failure. ....	45
3.3	Describing radios to WATSON. ....	47
3.4	Programming constructs. ....	48
 <b>4</b>	 <b>The Case of Q301</b>	 <b>51</b>
4.1	Lodging the complaint. ....	51
4.2	Signal tracing. ....	58
4.3	The loop problem solved. ....	64
4.4	The indictment of Q301. ....	69
 <b>5</b>	 <b>The Case of C422</b>	 <b>73</b>
5.1	Getting started. ....	73
5.2	Localizing to the AGC. ....	75
5.3	Indicting C422. ....	79
 <b>6</b>	 <b>A Case of Misalignment</b>	 <b>83</b>
6.1	The usual preliminaries. ....	83
6.2	Localizing the problem. ....	84
6.3	Alignment. ....	87
 <b>7</b>	 <b>The Case of Q407</b>	 <b>90</b>
7.1	Localizing to the audio driver amplifier. ....	90
7.2	Proposal and partial rationalization. ....	92
7.3	More DC rationalization. ....	97
7.4	AC rationalization. ....	99
 <b>8</b>	 <b>Bug Abstraction.</b>	 <b>100</b>
8.1	Q301. ....	100
8.2	C422. ....	106
8.3	The front-end. ....	112
8.4	Q407. ....	116
8.5	Using an abstraction. ....	116

<b>9</b>	<b>Signals</b>	<b>118</b>
9.1	Matching and propagation on the top-level plan-fragment. . . . .	119
9.2	Propagation and matching in the RF section/AF section cascade. . . . .	125
9.3	Propagation and matching in the RF section. . . . .	129
9.4	Signals as voltages and currents. . . . .	134
<b>10</b>	<b>Plans, Plan-fragments, and Designs</b>	<b>136</b>
10.1	Assembling the GR-78 design. . . . .	136
10.2	Goals, purposes, and rules. . . . .	139
10.3	Compiling determiners and agents. . . . .	141
10.4	Needs and provisions. . . . .	143
10.5	Further expansion of the RF portion of the GR-78 design. . . . .	144
10.6	Compiling caveats. . . . .	146
10.7	Plans with optional uses. . . . .	149
10.8	A few words about components. . . . .	151
10.9	Miscellaneous theorems. . . . .	152
<b>11</b>	<b>Miscellaneous Details</b>	<b>154</b>
11.1	Scenario completeness. . . . .	154
11.2	Consistency monitoring. . . . .	156
11.3	LOOPS. . . . .	157
11.4	Plan flattening. . . . .	160
<b>12</b>	<b>Conclusions</b>	<b>162</b>
12.1	The underlying concepts. . . . .	162
12.2	The machinery embodying the concepts. . . . .	165
12.3	Some universals of problem-solving in deliberate systems. . . . .	169
12.4	WATSON's shortcomings. . . . .	175
12.5	Directions for further research. . . . .	179
	<b>Appendix: Index of Terms</b>	<b>180</b>
	<b>Bibliography</b>	<b>183</b>

## Table of Figures

1.1	The localization paradigm. . . . .	3
1.2	An overview of the recursive localization process. . . . .	5
1.3	A partial Winston network for a design. . . . .	9
1.4	Further interpretation of the recursive localization process. . . . .	12
2.1	The GR-78 circuit diagram. . . . .	19
2.2	An RF amplifier. . . . .	20
2.3	Flattened block diagram of the GR-78. . . . .	21
2.4	Top-level block diagram of the GR-78 with RF and AF sections. . . . .	22
2.5	An expanded RF section. . . . .	23
2.6	Converter expanded into an oscillator and mixer. . . . .	25
2.7	An expanded oscillator. . . . .	25
2.8	The oscillator circuit. . . . .	26
2.9	The DC subcircuit. . . . .	27
2.10	An expanded IF strip. . . . .	31
2.11	An expanded AGC2. . . . .	31
2.12	Circuit diagram of the low-pass filter. . . . .	32
2.13	An expanded RF amplifier. . . . .	35
2.14	An expanded AF section. . . . .	36
2.15	An expanded audio amplifier. . . . .	37
2.16	Circuit diagram underlying the audio amplifier. . . . .	38
3.1	Failure localization in WATSON. . . . .	44
4.1	The context structure at the time of validation. . . . .	55
4.2	A simple cascade. . . . .	62
4.3	A cascade with a simple part join. . . . .	62
4.4	A fork and join combined to give a loop. . . . .	63
4.5	LOCAL's process state. . . . .	65
4.6	Three elements forward and one back. . . . .	65
5.1	The circuit details of AGC2. . . . .	80
7.1	Expansion of the audio driver plan. . . . .	94
7.2	Double expansion of the audio power amplifier. . . . .	95
7.3	An expansion of Ampl. . . . .	96
7.4	An expansion of Amp2. . . . .	97
9.1	The ob hierarchy in the signal representation. . . . .	119
9.2	Basic control structure in SIG-MATCH. . . . .	121
9.3	A port commonly shared at two levels. . . . .	127
9.4	Flattening of the RF section modulo IF1. . . . .	132
12.1	The top-level plan of an automobile. . . . .	171
12.2	Expanded top-level plan. . . . .	172
12.3	An expanded power train . . . . .	172
12.4	An expanded engine. . . . .	174

*This empty page was substituted for a  
blank page in the original document.*

## 1 Introduction

What does it mean to understand how a machine works? How can such an understanding be used to diagnose and repair a machine when it is broken? In this report I shall be concerned with answering these questions in the context of diagnosis and repair of local failures -- failures whose ultimate etiology can be resolved to a single component, or distinguished collection of components -- in radio receiver circuits. A theory of what it means to understand a mechanism is realized in a design<sup>1</sup> for a program, called WATSON, which attempts to mimic the performance of a competent radio technician. WATSON is guided in the localization process by teleological and causal annotation associated with the design of a given radio receiver circuit.

### 1.1 Why radios?

The reader might well suggest that radio diagnosis seems quite simple; why study it? Is it not sufficient to have a table (possibly large) of underlying causes of failures, indexed by symptom<sup>2</sup>? The answer is negative for two reasons. In the first place, a different table would be required for each radio, hardly the acme of generality. The other serious defect is that the index

-----  
<sup>1</sup> WATSON is not presently a running program, though many of the features of his design have been incorporated in various programs at one time or another. This report will give detailed descriptions of how WATSON would go about isolating a number of non-trivial failures. The documentation of these successful "hand simulations" will presumably convince you that the design is realizable as a program.

<sup>2</sup> A symptom will turn out to be an association of a class of incorrect outputs with a class of correct inputs for a radio -- a 'class' being a generalization of a particular observation. Later when I reveal WATSON's internal representations for symptoms, you will see that indexing them is actually not so easy as it might seem.

is not unique -- different underlying causes lead to the same symptoms. As a practical matter, this latter defect underlies the observation that almost any noticeable failure in a radio results in a lack of audible output.

So things are not so simple after all. There are two possible patches to the basic table-look-up (TLU) strategy; but before we consider them, let's examine the paradigm that seems to underly any possible strategy for finding the bug. It is hypothesize and test (see figure 1.1). A failure localization strategy is an embedding of this paradigm in some control structure<sup>1</sup>. In the proposed TLU strategy, failure enumeration is limited by looking only at those failures known to have the observed symptoms. Testing consists of pulling the component (whose failure is alleged to be causing the problem) from the circuit and instrumenting it to see if it meets its intrinsic specifications. The non-uniqueness of the symptom index implies insufficient pruning of the search space of possible failures. Hence, on the average, many components will be pulled before the real culprit is discovered.

Another possible strategy, the "big crunch" (BC), is to make use of the physical theory of electronic machines embodied in Kirchoff's laws together with the voltage-current (VI) characteristics of individual components such as resistors, transistors, capacitors, and so forth. For each component on the radio's circuit diagram, a particular failure is proposed. This failure would change the VI characteristics of the component in a well-defined way. Testing the hypothesis would consist of solving the equations of motion for the circuit in the circumstances of the failure and matching the solution against the observed behavior of the circuit<sup>2</sup>. If the match succeeds the bug has been found. Whatever reservations one may have concerning the feasibility of making the necessary measurements, they are certainly overwhelmed by doubts about the feasibility of the necessary computations. A circuit of even moderate complexity would require an

-----  
<sup>1</sup> Newell and Simon [1976] give an account in their Turing lecture of the importance of the hypothesize and test paradigm.

<sup>2</sup> The observed behavior here is revealed by voltage-current measurements in all the various branches of the circuit.

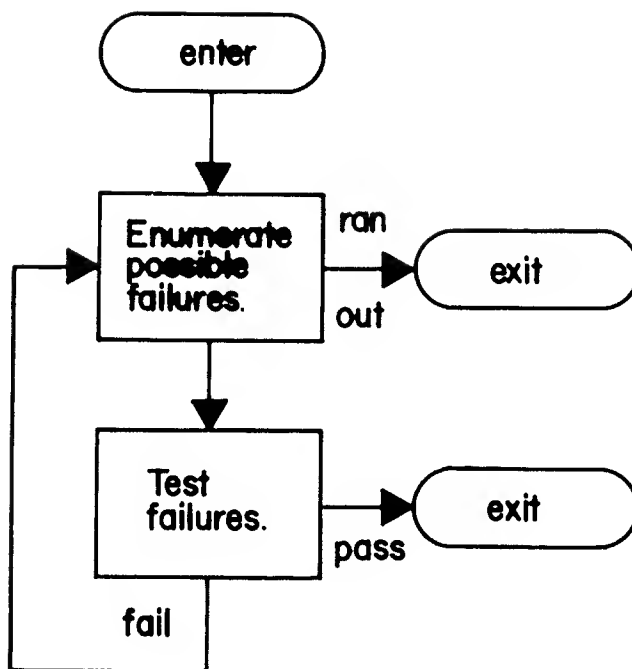


Figure 1.1 -- The localization paradigm.

astronomical amount of computation to carry out the BC strategy. Moreover, even when BC successfully uncovers the source of a bug, the answer is not very revealing. While BC will completely disambiguate failures that give rise to identical external symptoms, it provides so much detail that there is no way to formulate equivalence classes<sup>1</sup> of bugs.

Let's return to the problem of patching the TLU strategy. The problem is that when looking at the external symptomatology of the radio, we severely limit the available evidence. So it seems that we might ameliorate the problem of the many-to-one correspondence between bugs and symptoms by looking at some internal behavior of the radio as well. The question is, "What internal behavior?" The only distinguished sub-structures on the bare circuit diagram are components. Looking at symptomatology with respect to such components implies doing

-----  
<sup>1</sup> From a certain point of view, radios that differ at the circuit level will have identical abstract structures. Hence the different circuits will admit similar bug explanations modulo that viewpoint. This structural view gives rise to a partition of radio circuits according to the bugs they manifest.

precisely the same kinds of computations mentioned in the last paragraph. What seems necessary is the imposition of more macroscopic (and abstract) internal structures, like *amplifier*, *detector*, and *oscillator*, between the black-box view of the radio and its individual components. Instead of using external symptoms to suggest component failures, they might suggest *macro*-component failures. These macro-components could then have tables of underlying failures indexed by symptom. Adding another layer of symptom collection imposes a structure on the search space that can disambiguate the top-level symptoms.

Extending the notion of macro-component a little can solve the problem of generality as well. Consider that the BC strategy above presumes that every component can affect every other component. In reality designs are actually built up from modules wherein each component, more often than not, has negligible effect on components in other modules. Implicit in the radio circuit diagram is a hierarchical structure of modules. Many of these modules, like the amplifier, detector, and oscillator mentioned above, exist independently of a particular radio design. This suggests making explicit the hierarchical structure implied by the design, and specifying the interactions among the modules mentioned in that structure. Hypothesis formation could then be driven by the causal and teleological commentary associated with the various modules. Testing would consist of matching the behavior suggested by the causal commentary against the behavior that is actually observed. I am suggesting the strategy depicted in the flow-chart of figure 1.2 where the interpretation of the steps is:

1. Does the macro-component, MC, meet its specifications?
2. Propose a part, P, of MC that might be broken.
3. Apply the localization process to P which may itself be a macro-component.
4. Exit with the name of the failing part found at step 2.
5. Exit, complaining of false accusations.
6. Exit, complaining that there is no part to take the blame.

This is a variant of BC that need never ask questions about the insides of macro-components



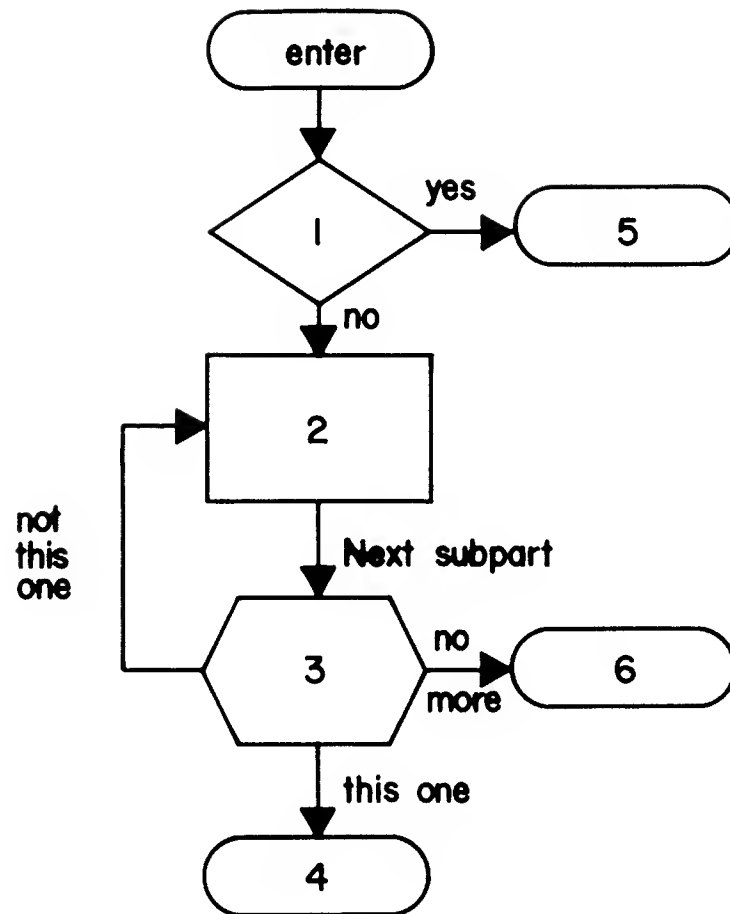


Figure 1.2 -- An overview of the recursive localization process.

unless there is some good reason to suspect them of causing trouble. Moreover, hierarchical structuring of BC allows schemes for proposal of failures to be as specifically tailored as is necessary to the particular kind of macro-component under investigation.

Radio repair offers a reasonable micro-world for understanding how machines work, because a general (feasible) strategy for carrying out such repairs requires the understanding of the function of the whole in terms of the functions of some collection of parts. In particular, WATSON's success at finding bugs in radios depends on his comprehension of design descriptions together with techniques of causal reasoning which are driven by, and manipulate those descriptions.

## 1.2 A theory of machine description.

When we say that we understand how a machine "works" we usually mean that we can decompose the machine into parts, describe how those parts interact, and indicate how those interactions lead to the *intended* behavior of the machine. To motivate the problem of representing mechanism, let us consider a first-order description of a machine familiar to everyone -- the automobile. Its *purpose* is to move a load between two places. To that end it has four wheels rigidly attached in parallel pairs to a chassis. The chassis supports the load. Also attached to the chassis is a motor that delivers a torque to an output shaft. The shaft is coupled to one parallel pair of wheels. The frictional interaction of the driving wheels with the road sets the entire machine in motion. Understanding this plan of an automobile presumes a great deal of physics knowledge in order to digest the notions of 'rigid attachment' and 'rotational coupling'. We will assume that knowledge to be implicit in a system for understanding mechanical devices. There are two ways we might expand the plan of the automobile: horizontally or vertically. In the former case we could add plan structure indicating how one parallel pair of wheels enables the car to be steered. This is an *extension* of the plan to account for additional function. An instance of vertical expansion would be a sub-plan detailing the parts of the motor, indicating how these parts interact to yield a torque at the output shaft. This is a *refinement* of the plan to give a more detailed explanation of some function.

Let's try to make an analogous description for a radio receiver. The purpose of the receiver is to select one of several possible signals from the "ether," demodulate it and transduce the modulation into sound. To that end a radio has two principle parts, a radio frequency (RF) section and an audio frequency (AF) section. The output of the RF section is coupled to the input of the AF section via a signal port. The RF section selects the signal of interest, demodulates it, and delivers the naked modulation signal to its output port. This modulation is processed by the AF section via amplification and frequency response equalization (e.g. RIAA, NAB, and various tone controls) to produce a signal suitable for conversion to sound by the loudspeaker. As with

an understander of automobile plans, we again assume that an understander of radio circuit plans has an implicit comprehension of the atomic concepts. Hence signal, signal coupling, and signal port are notions embedded in WATSON's procedures. A concept like "frequency selection" would be built up from more atomic concepts. Such a concept would be represented explicitly in the plan. As before, we can consider extensions and refinements of the radio plan. We might add another part to the plan, a power supply. Or we might consider a sub-plan for the RF section containing an RF amplifier, a converter, an intermediate frequency (IF) amplifier, and a detector, with a suitable explanation of how they interact to yield the overall function of selectivity and demodulation.

WATSON must be told more about a broken radio than its observed symptoms. To begin with he wants a complete circuit diagram of the radio. This diagram must then be *mapped* into a space of plans (which are types); that is, he needs its design. I call this mapping process *binding*. At each level in the binding process, parts are bound in the context<sup>1</sup> of bindings at higher levels. A top-level plan-fragment -- a token of a plan (perhaps like the one described in the last paragraph) -- is chosen. The radio is associated with this plan-fragment in an empty context. The plan, of which the plan-fragment is an instance, has a number of parts. Plan-fragments are chosen in turn for each of these parts. That is, each part is bound to its chosen plan-fragment in the context of the previous binding. This process is recursive. The product of this process is a tree of bindings. A branch of this tree is terminated by binding a part to a plan-fragment -- whose type might be the abstract resistor -- which cannot be decomposed further. A tree of bindings that has been *closed* in this manner is called a plan closure. Terminal elements in the tree may be associated with components on the radio's circuit diagram. The plan closure together with the component association and ancillary commentary is called the *design* of the radio. Plans

-----  
<sup>1</sup> A part may be viewed as a kind of variable which takes on a value in a design. Because the design induces a hierarchical structure on parts, the definitions of parts near the leaves of the hierarchy may freely refer to definitions near the root of the hierarchy, hence the usual scoping conventions of block-structured programming languages comes to mind.

distinguish interfaces of part interactions (such as ports and terminals) and the modality of interaction (such as signal and branch current). The input-output behavior of the parts, as viewed from their interfaces, is represented by an implicit plan sub-structure called an *io-contour*. The *io-contour* also indicates what overall effect results from the interactions of the plan's parts<sup>1</sup>. A plan may carry annotations concerning the external requirements to be met (or circumstances avoided) in order for it to perform its specified task. Finally, plans serve as the structures in terms of which failure mechanisms are abstracted. Keep in mind that since designs are built up from plan-fragments, the attributes of the types of the latter are inherited by the radios in whose designs their instantiations are bound. Figure 1.3 gives an illustrative relational network [Winston, 1975] of the kind that represents a design. INSTANCE-OF relations show how tokens of plans are given. A combination of PART-OF and INSTANTIATES-PART-IN relations yields a binding. Finally, the TOP-LEVEL-PLAN-FOR relation gives the name of a design and a plan-fragment used for its top-level description. One of the many descriptive features of designs not shown in this network is the fact that plan-fragments (the circular shapes in figure 1.3) can be identified with components on the circuit diagram.

### 1.3 Where do plans and designs come from?

Why must WATSON be told the design of the radio? Doesn't a good technician infer the design (from the circuit diagram) for himself? The circuit diagram of a radio typically supplies the technician with many hints about the radio's modular structure. Radio circuit diagrams have labels on them like "RF amp," "IF section," "IF can," "detector," etc. These labels denote goals which may be achieved by plans that are well known in the culture of radio design. The layouts of the circuit diagrams themselves are stylized so as to give certain kinds of information about the workings of the circuit, e.g. DC voltages (especially biases) tend to "fall" from top to bottom.

-----  
<sup>1</sup> The *io-contour* may be thought of as a distinguished sequence of verification conditions of the variety that appear in the Floyd-Hoare-style axiomatic semantics for programming languages.

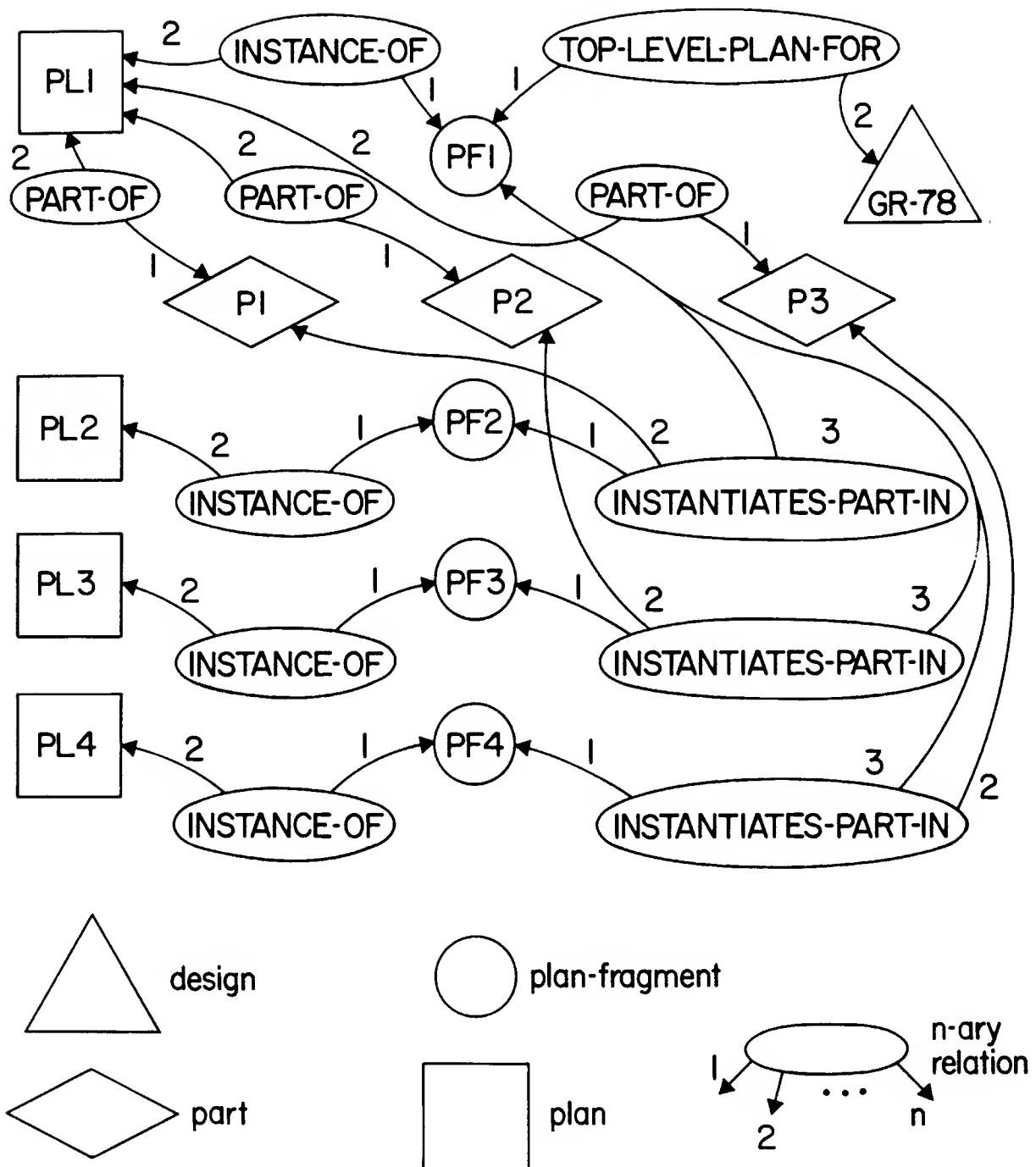


Figure 1.3 -- A partial Winston network for a design.

and signals "flow" from left to right. The parsing process that yields a design from a bare circuit diagram depends on knowing a great deal about the space of plans and the process of designing. Though a good technician has typically mastered the parsing problem, his skill is more akin to design than it is to debugging. Besides, the designer had the design in his head in the first place. The circuit diagram is simply a rather poor documentation of that design<sup>1</sup>. Let me reiterate that one of the aims of this research is to give a method of describing of how any radio works. As suggested by McCarthy [1968], a system that cannot be told such things is unlikely to be able to deduce them. The effectiveness of the descriptions will be measured by their success at facilitating the prediction of a radio's behavior and the guidance those descriptions give to the failure localization process.

Having decided to tell WATSON the design, a bit more needs to be said concerning the content of a design. As I have already explained, a design is a certain hierarchical identification of plans with abstract parts, or possibly components on a circuit diagram. There is in fact considerably more identifying structure. For example, the plan for an IF amplifier has an abstract parameter called its "mid-band." A plan for an IF amplifier will typically have a part called an input filter. This filter will have a parameter called "band-pass." The design identifies these two parameters as having, in some sense, the same value. Other identifications include abstract ports with nodes on the circuit diagram, abstract controls with various knobs and switches, named terminals with nodes on the circuit diagram, etc. The structure that started out as a tree actually supports considerable numbers of vines, bushes, ferns, and other flora that I shall reveal in chapters to come.

#### 1.4 A theory of machine diagnosis.

-----  
<sup>1</sup> One of the reasons for the poor documentation is the lack of a good formal language in which to do the documentation. WATSON proposes, among other things, a solution to this documentation problem.

The localization of failures, apart from being an interesting problem in its own right, serves as a test of whether WATSON has truly understood the machines that have been explained to him. WATSON's methods of failure localization, as suggested earlier, are applied recursively to the plan-fragments bound in the design of the failing radio. The flow-chart of figure 1.4 shows the recursive control structure of the failure localizing process in greater detail. Starting with the plan-fragment (referring to figure 1.3 may be helpful) bound at the top-level to the radio, he checks his notes to see if he has previously abstracted any bugs (for that plan-fragment's type) whose symptom would match the observed symptom<sup>1</sup>. The abstracted bug, among other things, associates a symptom for the plan, a part of the plan, and a *sign* for that part. The sign is a symptom for the part. In order to put the blame on that part, the part must exhibit the correct sign. If the sign is present, the localization is recursively activated by dredging up the plan to which that part is bound, with the sign becoming the symptom at the new level of recursion.

If the indicated sign is not present, there may be other bugs known to WATSON whose patterns of symptomatic behavior make them applicable in the present situation. WATSON exhausts all such candidates before proceeding to other hypothesis-generation strategies. One such strategy, LS2, relies on the essentially causal nature of the machines under consideration. Plans may give the structure of the *flow* processes being carried out in the machines they explain. Such processes can be *back-traced*<sup>2</sup>. That is, starting at the final output, and knowing what process each part is designed to accomplish, WATSON can work his way along the processing path(s) until he finds a part whose output is incorrect given its input(s). If radio plans were strictly causal, back-tracing would always lead directly to localizing the failure. Unfortunately, plans have parts which are not really *unilateral*<sup>3</sup>. Moreover, the flow paths in plans may have

<sup>1</sup> Looking to see if the answer is already known is one of several possible failure localization methods or strategies. I shall refer to it as the "LS1" strategy.

<sup>2</sup> Note that implicit in the back-tracing process is the assumption of (among other things) the independence of parts in different modules, about which I remarked earlier.

<sup>3</sup> A part in a flow process is unilateral if its output side cannot affect its input side. Indeed, excluded from this class are various passive filter networks.

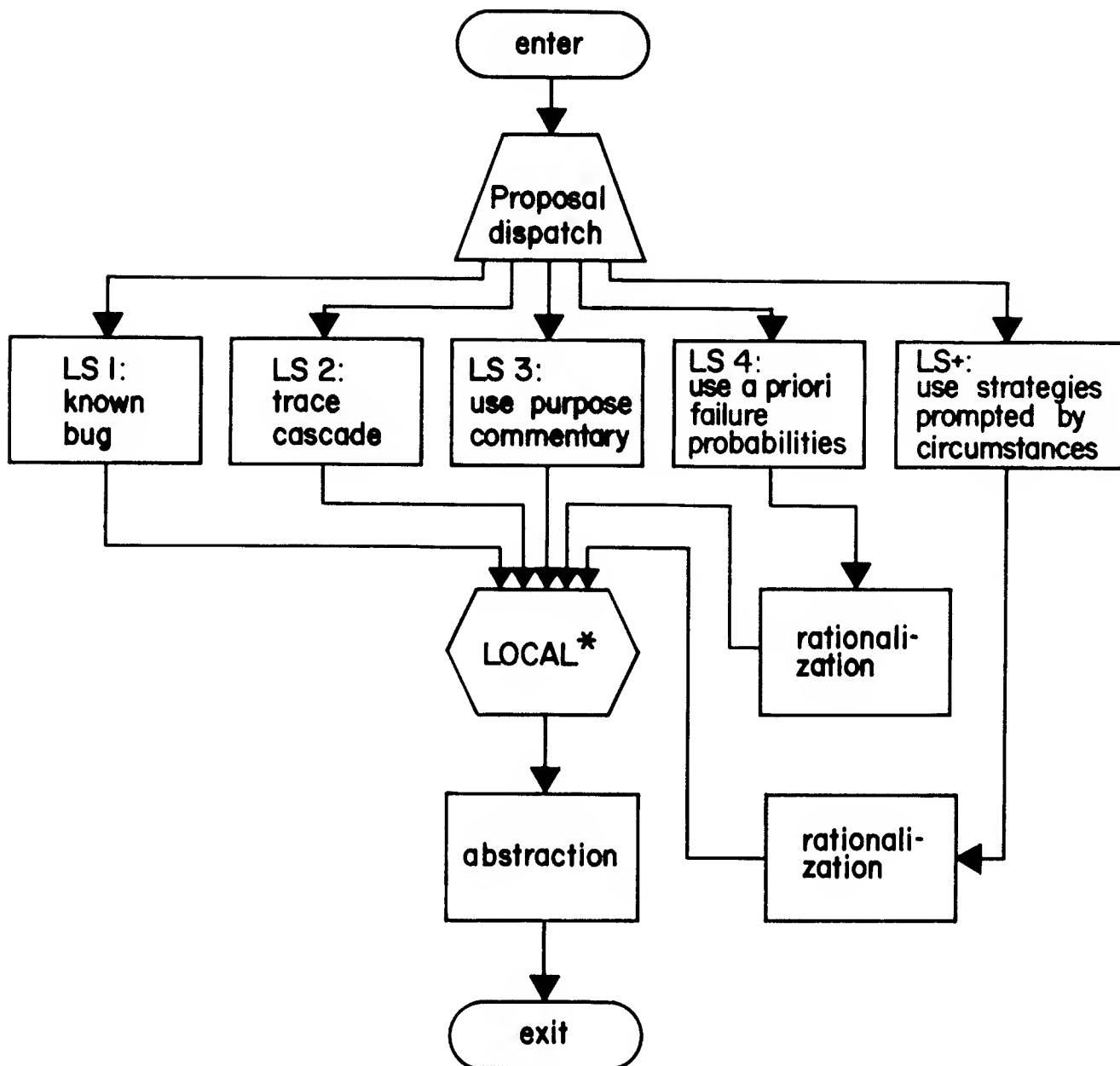


Figure 1.4 -- Further interpretation of the recursive localization process.

feedback. Such non-idealities necessitate other hypothesis-generation mechanisms.

The next method, LS3 , works by considering the purposes of various parts in a plan.



A *purpose* is represented by a predicate whose truth is guaranteed by a *rule*. Networks of these rules and predicates make up the io-contour mentioned earlier. The io-contour, then, constitutes a kind of electronic calculus that plays a role at the module level analogous to that played by differential equations of motion at the component level. Forming the hypothesis consists of proposing that some predicate is no longer true (i.e. its supporting rule has been invalidated) and predicting the consequences of such a turn of events. Another hypothesis-generation mechanism, LS4, considers the circuit components associated with the terminal plan parts and suggests failures in those components based on the *a priori* probabilities of the components' succumbing to known faults. LS3 and LS4 have the property that the hypotheses they produce must be *rationalized*. Unlike back-tracing and hypothesizing previously abstracted bugs, these other mechanisms can hypothesize failures that do not necessarily lead to the observed symptom. Hence the consequences of the hypothesis must be deduced and shown to agree with the observed symptom. When the hypothesis-generating mechanisms succeed in producing a plausibly failing part, WATSON abstracts the failure mechanism (if it is previously unknown) and invokes the localization process recursively. The recursion terminates at such time as a localized part is found to be associated with a circuit component and *verified* by removing the component and examining it for the proposed failure. It goes without saying that at any level of localization contradictory evidence may be discovered that necessitates backing out of an hypothesis<sup>1</sup>.

### 1.5 Related work.

There are a number of programs whose motivations or methodologies are close enough to WATSON's to warrant comparison and contrast. They fall into two broad categories: programs whose principal concerns are debugging, diagnosis or "linear" problem-solving; and programs

-----  
<sup>1</sup> There are also localization strategies that are specific to particular plans. Such strategies will be referred to collectively as LS+. The notion here is that there are tricks that frequently work in radio repair (like inspecting the heater filaments in tube radios) that are not sufficiently universal to be subsumed by bug abstraction.

that are essentially computer-aided analysis systems charged with yielding predictions about the dynamic and/or static behavior of particular electronic circuits. The first category includes Sussman's HACKER [1975], Goldstein's MYCROFT [1974], Sacerdoti's NOAH [1975], Shortliffe's MYCIN [1974 and 1976], Brown's SOPHIE [Brown et al., 1974, Brown et al., 1975], and de Kleer's troubleshooting program [1975]. The second category includes Penfield's MARTHA [1971], Dertouzos' CIRCAL [1967], and EL, a recent effort of Sussman and Stallman [1975], and Roylance [1975]. WATSON's immediate ancestors are HACKER and MYCROFT. HACKER is a program that becomes increasingly skilled at solving blocks-stacking problems. He does so by debugging old programs to fit new situations. WATSON, in contrast, becomes increasingly skilled at debugging radios. He does so by enlarging his repertoire of bug classifications. Since the bug classifications are abstracted in terms of plans which may be bound in many radio designs, the new knowledge is available in any of those designs. HACKER tries to explain bugs in terms of pre-compiled classes. The success of both programs is attributable to the extensive commentary on the machines of their respective domains. This commentary indicates what the parts are trying to accomplish and for whom. MYCROFT is an expert at debugging a limited class of LOGO programs. The class of programs in question has a particularly structured product -- pictures. Goldstein shows how models of the intended product, together with the user written program, can be used to infer the user's plan for his program. Since bugs generally arise at the interfaces of plan steps, knowledge of potential bad interactions among those steps allows MYCROFT to repair the plan, and thence the program. He can also repair some programs that are not correct implementations of a plan. WATSON similarly relies on the causal and teleological commentary embedded in plans to guide the debugging process. His job, however, is to convert a perfect plan into one that would exhibit the observed symptoms. The fundamental advance of WATSON over HACKER and MYCROFT is two-fold. In the first place, WATSON has a qualitative theory of how the machines of interest to him work. He can make predictions about their behavior by means other than running them. Or more accurately, he models machine behavior at various levels of detail. In the second place, he

exercises hierarchical control of local reasoning. This keeps him out of the trouble that theorem provers classically encounter. No more facts pop into his head than are relevant at the time. As it turns out, the phenomenon of locality will be the key ingredient in both the representational mechanisms and control structures embodied by WATSON.

NOAH is a producer of plans while WATSON is mostly a user (though sometimes incremental modifier) of plans. Despite their different goals, they both rely on hierarchical plan representations. Sacerdoti's procedural net has many of the same attributes as my space of plans. His nodes (my plans) represent various levels of detail in actions (processing) used to modify the environment. Both systems apply various evaluators to substructures of their plans to compute what effect a plan has on its environment. The differences in our plan structures seem to arise mostly from his interest in manipulating the *temporal* structure of plans in contrast to my interest in manipulating their *action* structure.

Shortliffe's system exhibits a number of design choices similar to those I have made in WATSON. My symbolic description of signals is not unlike his parameterized descriptions of patients. His system relies on a system of rules to deduce correct therapies. I have a system of rules for matching signals and branch variables (currents and voltages), and a system of rules for propagating signals and branch variables to do causal reasoning. More importantly, MYCIN's and WATSON's uses of rules differ in that in the former system, the rules are the diagnosis whereas in the latter rules are used as predictive aids in guiding the diagnostic process. WATSON attempts to embody a general class of diagnostic techniques to be applied to particular machine descriptions, whereas MYCIN makes use of a collection of diagnostic techniques directed specifically at the artifact being debugged. It should also be reiterated that WATSON's hierarchical representational scheme allows rules to be understood in terms of (presumably) more primitive rules. Essentially MYCIN has no theory other than the rules of diagnosis. If a physiological theory were added, the resulting system would be akin to WATSON.

Since SOPHIE is a system for the teaching of trouble-shooting in electronic instruments,

It is reasonable to expect that she must confront many of the same technical issues as confronted by WATSON. Her very different solutions to the problems seem to stem from a number of factors: desire for speed, general question answering ability, concentration on one particular instrument, and the fact that that instrument is a power supply<sup>1</sup>. SOPHIE's deductive methodology seems to rely on converting essentially qualitative questions to quantitative ones that can be posed to a simulator, augmented with specific knowledge of a particular power supply. The representational machinery is geared to supporting that interface and to lending semantic support to a natural language front end.

De Kleer is concerned, as I am, with keeping the trouble-shooting program's reasoning confined to locally available knowledge. He restricts his qualitative methods to DC circuitry, however. This is partly because AC qualitative analysis seems to require hierarchical, teleological structures (which he does not investigate), and because he is interested in optimal measurement strategies and the purposes that underlie measurements.

Sussman, Stallman, and Roylance have recently reported on a new circuit analysis program called EL. EL makes use of causal reasoning in much the same local fashion as does WATSON. Since EL's plan-fragments have no "insides," EL cannot reap the benefits of hierarchical reasoning. This means, of course, that EL's notion of causality is "flat," a circumstance that limits the complexity of the mechanisms to be analyzed as well as the depth of their analysis.

MARTHA and CIRCAL are classical analytic aids to circuit design. The goals underlying their designs are quite different from those underlying WATSON's. They embody a great deal of knowledge about analytic models of electronic circuits. While WATSON finds it straightforward to summarize a radio's qualitative behavior at various levels of detail (a synthetic problem), MARTHA's and CIRCAL's major stumbling blocks arise when trying to extract what they know in terms of the designer's problem. Conversely WATSON cannot explain a fourteen-pole Butterworth filter

-----  
<sup>1</sup> A power supply is a machine that is in a sense designed to exhibit no dynamic behavior. Hence its behavior is easily characterized by numbers when operating correctly.

(because he has no theory of interaction for the filter's parts), while the analytic systems find detailed mathematical explanations (Remember the BC strategy!) of such devices straightforward. Clearly what is needed is an engineer's aid that is a synthesis of the two kinds of systems.

#### 1.6 What is to come.

The remainder of this report hinges heavily on the next chapter. It gives an informal account of how WATSON goes about his business. Beyond that, chapters are organized more or less breadth-first. I attempt to give appropriate forward pointers so that you may examine any aspect of the design in as much detail as suits your fancy ...

## **2 Scenarios of WATSON's Performance**

In this chapter I shall present four scenarios of WATSON's execution of the trouble-shooting task. These scenarios will be informal in two senses. First, nothing will be said for now about how we might communicate to WATSON the nature of the faulty radio's symptoms or the radio's design. The second sense of informality lies in leaving unrevealed for the time being the precise representations of WATSON's knowledge and deductive methods. The scenarios will be presented as if they were protocols of a human technician. The purpose of this chapter is, therefore, to acquaint you with what WATSON can do, rather than how he does it.

Each scenario is concerned with a local failure in a particular radio receiver, the Heathkit® GR-78 [Heath Company, 1969] whose circuit diagram is presented in figure 2.1. Although the receiver actually covers six bands between 190 kHz and 30 MHz, and is capable of amplitude modulated (AM), continuous wave (CW), or single sideband (SSB) reception, WATSON's model of the radio is that it is an AM receiver for the band between 3 MHz and 7.5 MHz. This simplification entails no loss of generality since WATSON already has mechanisms for representing variability in the configuration of the radio (see chapter 10). These mechanisms could readily handle a more complete model of the GR-78, including switching bands and changing the mode of demodulation. Moreover, the representations and methodologies do not depend on a particular receiver. The last observation addresses a more important question that might be raised concerning the use of a single instrument as a source of examples. Might not this indicate the WATSON's methods are limited to that instrument? The GR-78 was chosen in the first place because it is a receiver of moderate complexity. As such it incorporates in its design a substantial set of plans from the space of possible radio circuit plans. Also the methods employed by WATSON have been hand-simulated in the trouble-shooting of circuits other than the GR-78.

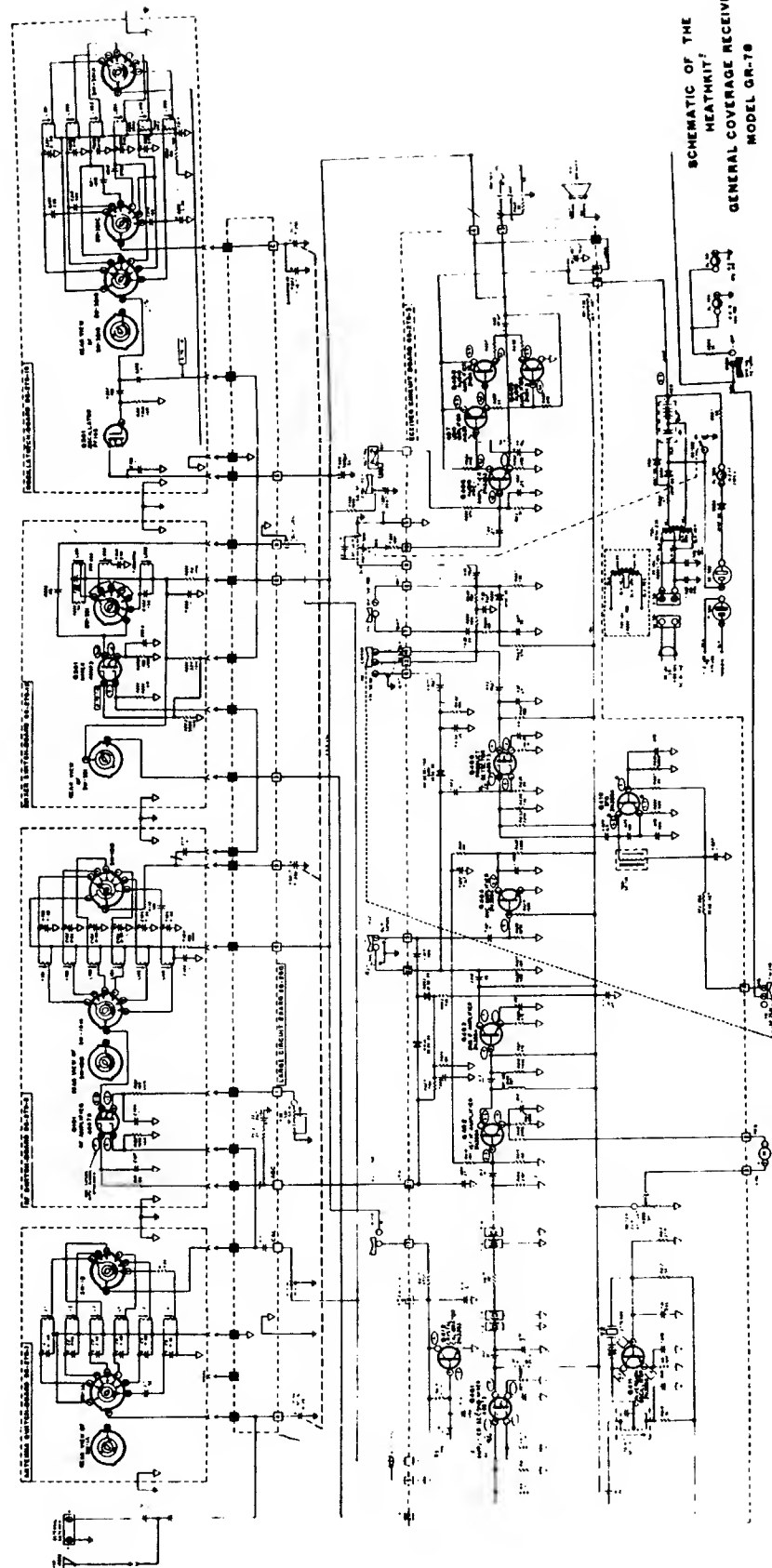


Figure 2.1 -- The GR-78 circuit diagram.

Before getting on with the scenarios we must establish some temporary conventions for thinking about radio plans. We have already made use of standard circuit diagram notation in figure 2.1. We will continue to do so. Since the circuit diagram is rather unwieldy in its entirety, we will often consider pieces of it as in figure 2.2. Another thinking aid we will employ is the block diagram as in figure 2.3. Such diagrams consist of functional blocks connected by links representing the signal flow. The heads and tails of these links are ports which we will sometimes identify with pairs of nodes on the (a sub-)circuit diagram of the GR-78. We will eventually see how these two partial plan representations (block diagrams and circuit diagrams) are formally realized in WATSON's data structures.

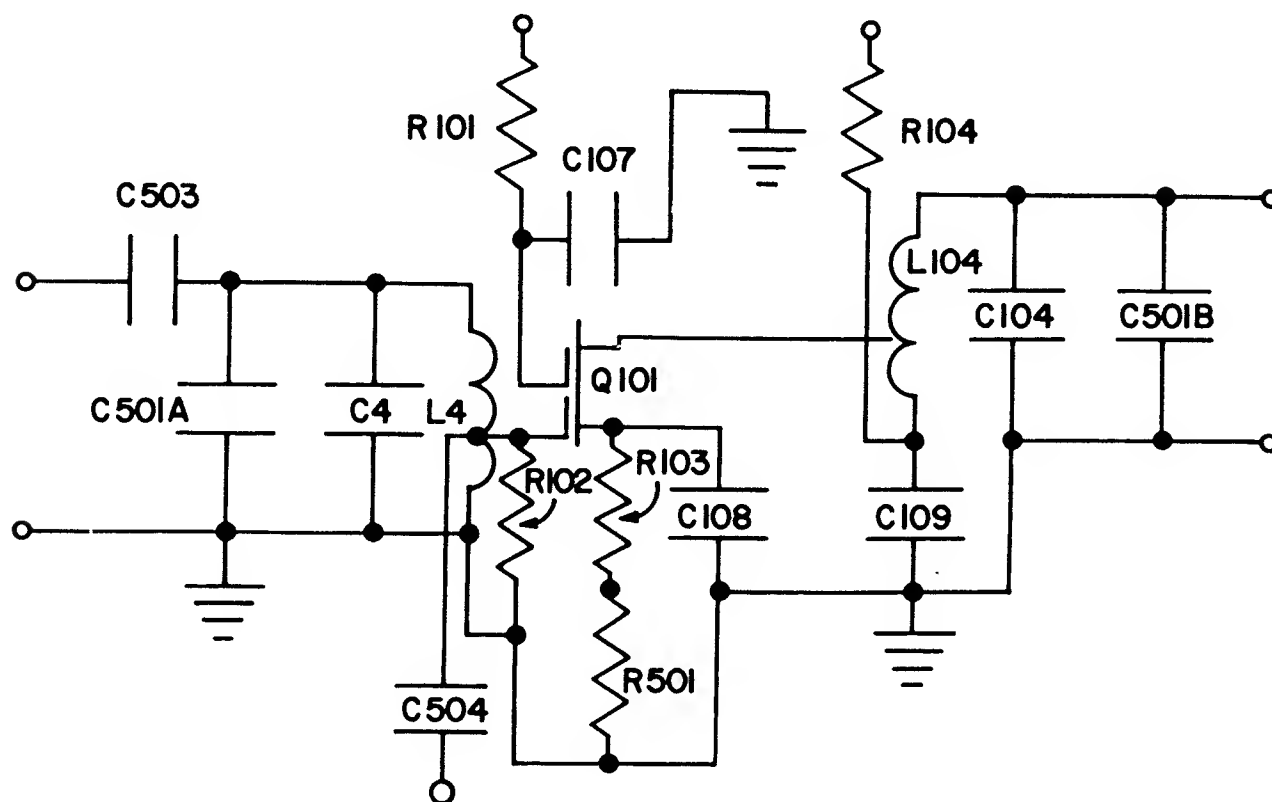
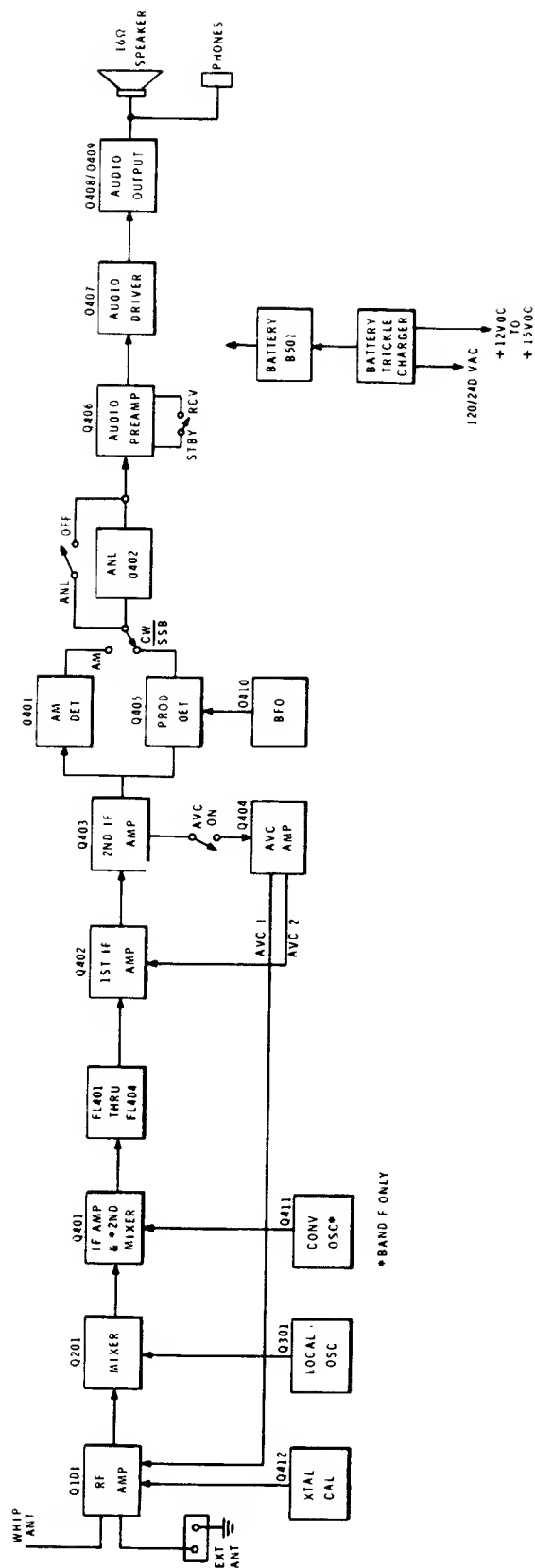


Figure 2.2 -- An RF amplifier.

2.1 A opened junction in a transistor.





BLOCK DIAGRAM

Figure 2.3 -- Flattened block diagram of the GR-78.

WATSON is presented with a GR-78 which seems to have no audio output under any circumstances<sup>1</sup>. WATSON knows that the radio is organized as in figure 2.4. He checks to see that the output of the AF section is invariant with station selection and volume-control setting. This is indeed the case. He is now ready to do trouble-shooting using a signal that is completely under his control. He calls for a signal generator to be connected at the antenna side of the RF section<sup>2</sup>, and requests that it be set to generate a 5 MHz signal, amplitude modulated at 1 kHz and tunes the GR-78 around 5 MHz. Still no audio output. In particular, there is no 1 kHz output signal.

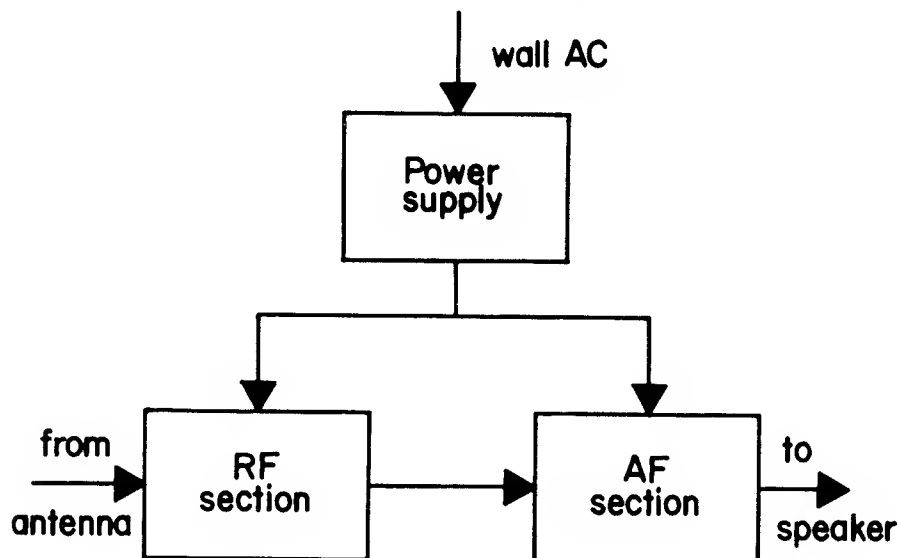


Figure 2.4 -- Top-level block diagram of the GR-78 with RF and AF sections.

Now he is ready to do back-tracing. What does the input signal to the AF section look

-----  
<sup>1</sup> This observation ignores the thermal noise that will appear if the radio's volume control is turned up sufficiently. The presence of such noise could, in fact, be used as a hint by WATSON the power supply is likely to be working. The procedural complication is not particularly revealing in the present circumstances. The discussion of caveats in later chapters shows how such hints might be formalized and used in plan-specific localization strategies.

<sup>2</sup> Notice that I have not mentioned The AF section's input from the power supply -- a power port. In general I shall not mention back-tracing through power ports (unless they are the source of a problem), as checking them is monotonously straightforward. Assume, however, that in every recursive application of the localization process such ports are in fact being checked.

like? WATSON knows that there should be a 1 kHz signal there, but there is not. So the trouble is probably not in the AF section, and therefore must (discounting the possibility of a shorted input port<sup>1</sup>) originate somewhere before it. Hence the RF section should be examined next. WATSON knows that the RF section's input signal must be correct since the signal generator is supplying it. The trouble then is most probably in the RF section. WATSON must now examine the organization of the RF section, which is presented in figure 2.5.

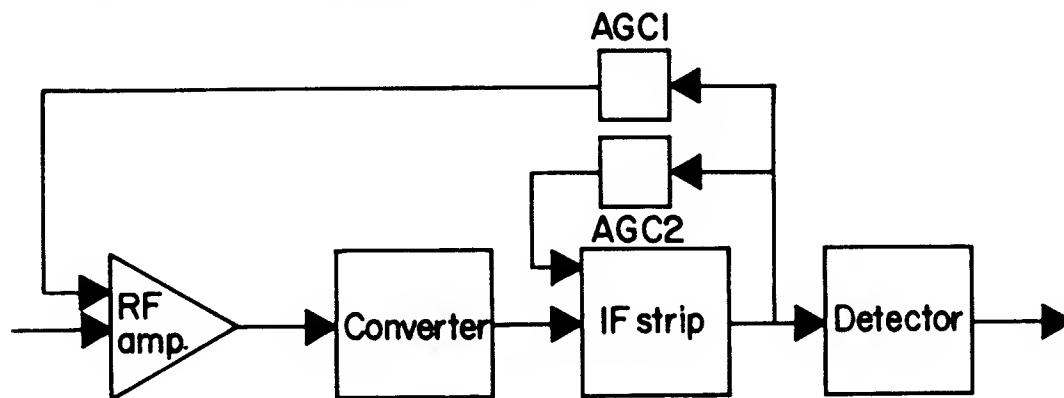


Figure 2.5 -- An expanded RF section.

Since the output port of the detector is the same as the output port of the RF section, the signal at that port is already known to be bad (i.e. not a 1 kHz signal). Looking at the input port of the detector, WATSON discovers that it is not a 455 kHz signal amplitude modulated at 1 kHz, as the operational theory of this radio predicts that it should be. The preceding module is the IF section, whose functional semantics demand that its input too should be a 455 kHz AM signal (and perhaps some spurs), though having a smaller amplitude than its corresponding output. Theory does not correspond to reality in this case. Back-tracing again, WATSON arrives at the converter. The theory of the GR-78 indicates that the input signal should consist mostly of a 5 MHz component, amplitude modulated at 1 kHz. Inspection verifies this<sup>2</sup>.

-----  
<sup>1</sup> This possibility is "pushed." If the possibility of "before AF section" fails to pan out for some reason, WATSON will return to this point.

<sup>2</sup> Note that this last measurement is difficult to do without the aid of a very elegant RF measuring instrument (which turns out to be another receiver in essence). The control structure employed by most technicians would probably hypothesize that the input to the converter was, in

WATSON now has his hands on a module whose inputs seem correct, but whose output is not. It is a good bet that the trouble is inside -- which calls for expanding the inside. Figure 2.6 shows the main ingredients of a converter as seen by WATSON. The mixer has two signal inputs. WATSON has already verified that one of these inputs is correct. The other should be an unmodulated sinusoid of frequency 455 kHz plus the frequency of the broadcast signal -- 5.455 MHz in this case. It is not. Since the oscillator has no inputs, it is very likely the culprit. Figure 2.7 shows an expanded oscillator. WATSON continues the tracing process at this level of detail. He already knows that the output of the tank is wrong. Examining its input reveals no oscillation there either. Further investigation shows oscillation neither at the signal input of the amplifier nor the input of the amplitude stabilizer<sup>1</sup>. Notice that WATSON has just completed a loop in the course of tracing the signal. He has not been able to localize the failure further because all of the modules have correct outputs given their inputs.

This situation calls for the invocation of WATSON's loop specialist. One technique that can be used in such situations is to break the loop and independently supply the signal that should be present at the break point. Ordinary tracing can then be used to find the module that is causing the problem. A variant on this technique involves the injection of a signal at some point in the loop without making a physical break. WATSON can then check to see whether the signal propagates forward correctly. Loop breaking techniques are the loop specialist's *forte*, but they are expensive to use. So WATSON first tries for a "quick and dirty" solution using an *ad hoc* technique.

-----  
fact, okay and continue. The holding of some hypotheses in abeyance while pursuing other hypotheses in parallel is easily achieved using CONNIVER's generalized control structure. For the purposes of the present exposition, we shall presume that WATSON will always choose to motivate hypotheses by measurements rather than optimistically assuming that a hypothesis will work out in the absence of such motivation.

<sup>1</sup> The sense of 'input' here is purely abstract. Standing alone, the RC network underlying the amplitude stabilizer has no input or output. Viewed as a two-port network it is quite symmetric. Its use in the oscillator suggests an input side and an output side.

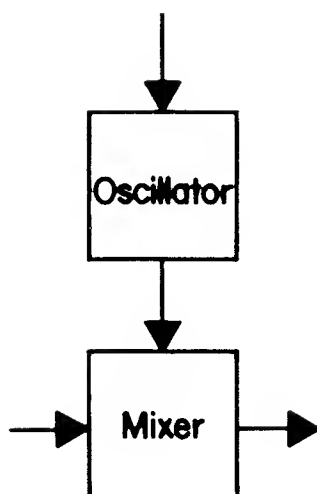


Figure 2.6 -- Converter expanded into an oscillator and mixer.

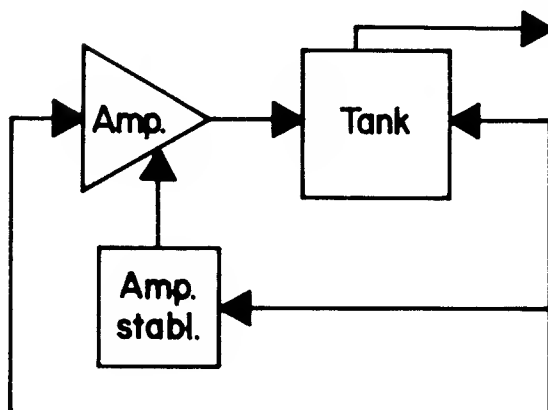


Figure 2.7 -- An expanded oscillator.

The heuristic used is to consider each of the active modules<sup>1</sup> in the block diagram. These modules get expanded to the level of detail of the circuit diagram where the consistency of the bias voltages may be examined. A quick check in the present case reveals that the gate bias of the junction field effect transistor (JFET) Q301 in figure 2.8 is inconsistent with the source

-----  
<sup>1</sup> An active module is one with power ports in addition to signal ports. The amplifier in figure 2.7 actually has a power port associated with it.

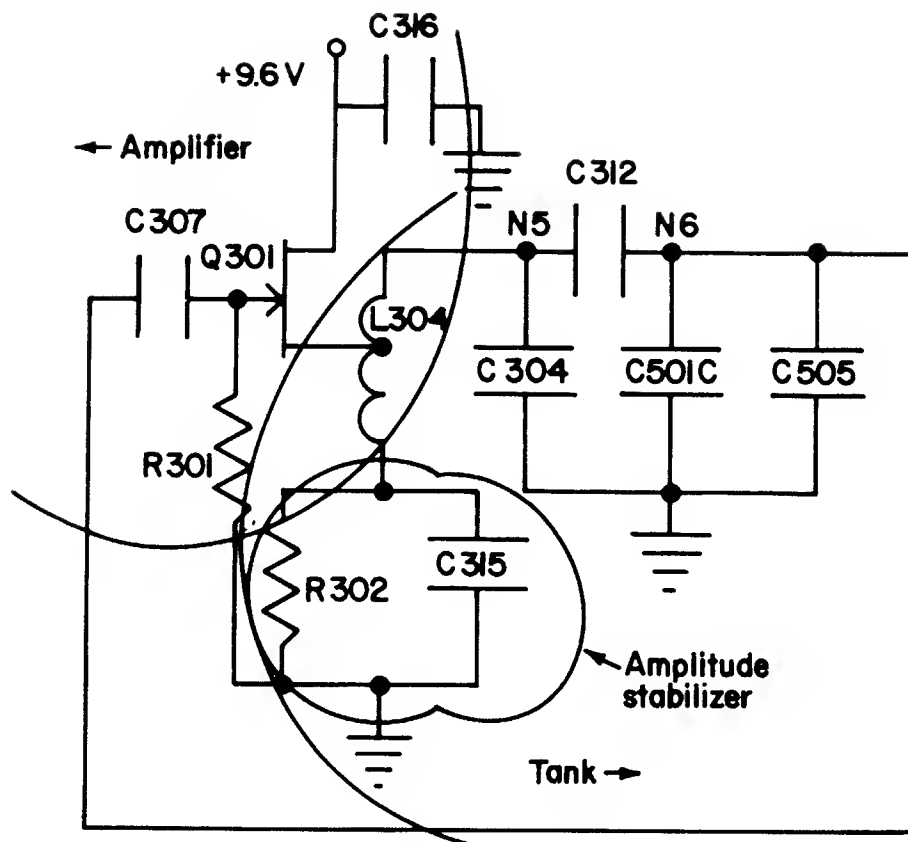


Figure 2.8 -- The oscillator circuit.

and drain biases<sup>1</sup>. What could cause this situation? Remember that WATSON is now looking at the amplifier at the circuit diagram level. Unlike the situation in plans represented by block diagrams, the causal relationships among the parts are not so clear. Knowing nothing else about a particular circuit, WATSON looks for possible failures in the active components first. One such possibility is the opening of the p-n junction at the gate of the JFET. Analysis indicates that the

-----  
<sup>1</sup> If the oscillator were operating, the bias at the gate would be considerably different from the quiescent bias. The mechanism that supports the operating bias scheme is quite difficult to deduce because it depends on the dynamic non-linearity of the JFET. Thus it seems appropriate to represent it explicitly. If it is quiescent, the JFET is biased for class A operation. This enables the oscillator to start up. Figure 2.9 is the relevant subcircuit for this bias scheme. Commentary on the oscillator and its amplifier mentions both bias schemes.

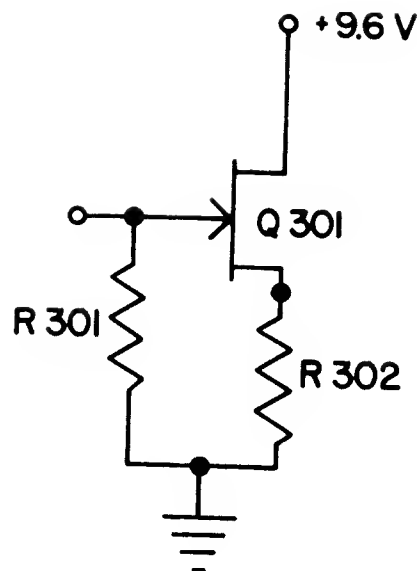


Figure 2.9 – The DC subcircuit.

observed drain current would be larger than the drain current expected (when operating faultlessly), given the observed gate-source voltage. An opened gate implies zero gate current, hence the voltage drop across R302 must be zero. These DC consequences predicted by WATSON (in the context of class A operation) agree with reality.

What about the AC consequences? WATSON reasons that a displacement up or down from the gate bias voltage would have no effect on the output side of the transistor given the rules governing Q301's behavior in the opened gate situation. How does this AC symptomatology at the circuit diagram level *lift* to the block diagram of figure 2.7? WATSON knows that the purpose of the feed-back loop is to stabilize the output amplitude of the oscillator. Would the oscillator remain stable if the amplifier were not doing its job of providing AC gain? The answer is no, for WATSON reasons as follows: Suppose the amplitude at the output of the tank were to decrease. To restore the output to the desired amplitude, the amplifier must kick the tuned circuit that underlies the tank module. But if the JFET is actually failing in the mode proposed, there is no way to make the JFET's gain be non-zero. Eventually all of the oscillatory energy will be dissipated and the oscillator will become quiescent as observed. Notice that the driving force

in WATSON's reasoning comes from considering the teleology of the stability arising from the feedback loop. WATSON pulls the JFET for inspection and discovers that it is indeed opened at the gate<sup>1</sup>.

Having successfully isolated the failing component, WATSON abstracts the bug he has discovered. The abstraction occurs in the "returning" phase of the recursive localization process. For each plan along the localization path, he may record the input-output symptomatology that caused him to visit the plan, the input-output symptomatology (the sign) of the part of the plan which was discovered to be causing the trouble, and the mechanism by which the failure of the part causes the observed behavior<sup>2</sup>. At the level of figure 2.9 the symptom is that there is no AC gain, the sign is an unusually low gate to source bias for class A operation. The mechanism is that incremental displacements at the gate cause nothing to happen at the source. At the level of figure 2.7 the symptom is that there is no oscillatory output, and the sign is that there is no incremental gain<sup>3</sup>. The mechanism has two parts. If the circuit were ever oscillating, it would have ceased to do so owing to dissipation in the passive modules which is not offset by the active module. If the circuit never oscillated, it will never start since doing so depends on having ambient thermal noise amplified by the amplifier of the oscillator. For figure 2.6 the symptom is that the expected output mentioned above is not there. The sign is that the oscillator is not oscillating. The mechanism is implicit in the "tuned" nature of the mixer: mixing the input signal

-----  
<sup>1</sup> I should also point out that we have only seen WATSON's success paths. At each level of hypothesis generation, there may be more than one plausible hypothesis. WATSON attempts to eliminate as many hypotheses as he can in the rationalization process. It is possible, nonetheless, that several components may be equally plausible loci of failure. In this case the real culprit can only be determined by removing the components in question from the circuit and measuring their intrinsic properties.

<sup>2</sup> Certain criteria are applied to decide whether or not a bug is worth remembering. The nature of the criteria will not be understandable until considerably more of WATSON's machinery is displayed. For now, we shall imagine that an abstraction occurs for every level of localization. As it turns out, the description of the bug mechanism will be intimately related to the notion of a part's purpose in a plan.

<sup>3</sup> Note that there may be no reliable way of checking for this sign without breaking the loop that is intrinsic to the oscillator.



with the oscillator signal that is not 455 kHz offset by the broadcast frequency will cause the broadcast signal to be swallowed up, just as ought to be the case with stations to which the radio is not tuned. By composition, the converter fails to pass the input signal as well. For figure 2.5 the symptom is that there is no output, the sign of the converter is that there is no output, and the mechanism is the one that follows directly from the composition of parts: the lack of output from a part of a flow process leads to a lack of output from the overall process.

## 2.2 A shorted capacitor.

In this scenario, the presenting symptom of the GR-78 is that it has no audible output when tuned to weak stations. Very strong stations are barely audible and heavily distorted. Again WATSON refers to the plan described in figure 2.4. He requests that a signal generator be set up to generate a 5 MHz carrier of moderate strength modulated at 1 kHz while the radio's volume control is set to a normal listening level. The radio is tuned to 5 MHz. He notes the lack of audible output and asks for an increase in signal strength until an audible output and distortion appear. WATSON performs similar experiments at 0.1 MHz intervals over the band. At each "station" the same symptomatology may be induced, and remains present independent of volume control setting.

With the signal generator producing a 5 MHz signal, modulated at 1 kHz, and of sufficient strength to induce distortion, WATSON begins the back-trace. Knowing that the output of the AF section is wrong, he looks at its input and discovers that it too is distorted. Since the input to the RF section is known to be correct, the RF section is conjectured to be the site of the problem. Expanding the RF section into a more detailed plan (again represented by figure 2.5) he quickly discovers that the output of the IF strip is distorted, but its input (from the converter) is not<sup>1</sup>.

-----  
<sup>1</sup> Again this is a difficult RF measurement requiring both a sensitive RF probe and a frequency analyzing instrument. A not so well equipped technician would hypothesize the reasonableness of the IF strip's input from the converter, and continue.

WATSON is not yet ready to blame the trouble on the IF strip. Note that there is an auxiliary input from AGC2 to the IF strip. In fact, the IF strip and AGC2 together form a feedback loop whose teleology is known to WATSON. The loop's purpose is to stabilize the strength of the IF strip's output signal. WATSON's theory of mechanism for this loop is that the output signal strength's rising above the desired value causes the gain of the IF strip to be decreased. Similarly if the output signal strength falls below the desired level, the gain of the IF strip increases. Another feature of the mechanism is that it has an associated time constant. That is, the description of the mechanism explicitly includes the notion that the strength is determined by time-averaging over a known interval. (The formal details of this description are elucidated in section 5.3.)

In the present situation, an examination of the port that brings the signal from AGC2 to the IF strip reveals a bias of 0 V -- clearly incorrect. How does WATSON know this? He deduces it from the teleological commentary associated with the feedback loop together with information about the input signal strength obtained from measurement. Moreover there are caveats associated with the IF strip indicating minimum and maximum values that the bias at the automatic gain control port can take on. The bias at that port is at ground, hence well under the advertized minimum value. What effect does this have on the IF strip? As already mentioned, the IF strip's associated commentary explicitly prohibits its use with the observed biasing. Consequently to understand how the IF strip is affected by the adverse biasing, WATSON must look at its plan structure in greater detail. Forward reasoning on the plan described in figure 2.10 indicates that the output of the IF strip will be at best an amplified and harmonically distorted image of its input -- if the input signal is large.

The plan for AGC2 is illustrated in figure 2.11. Looking at the input of the low-pass filter (LO-PASS), WATSON finds an amplified and rectified version of AGC2's input. The gain of the AVC amplifier is, in fact, not as large as it should be. But back-tracing in the usual fashion, WATSON notices that the input and output of the low pass filter are not identical --

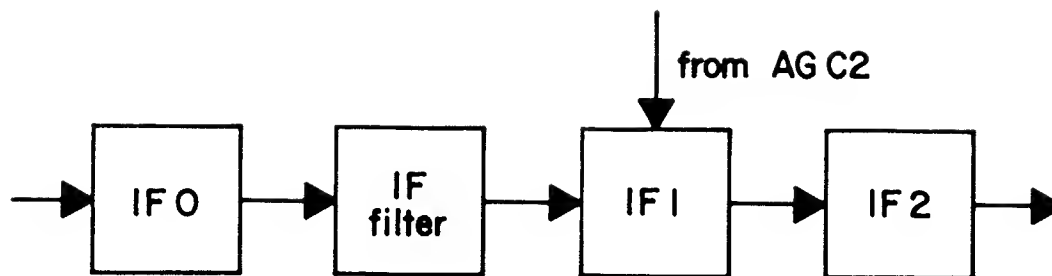


Figure 2.10 -- An expanded IF strip.

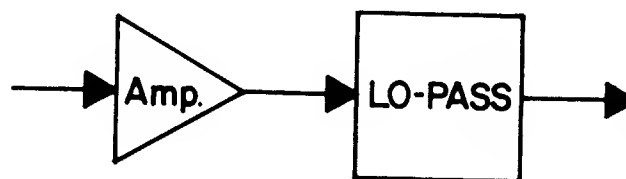


Figure 2.11 -- An expanded AGC2.

modulo a DC offset -- as prescribed. In fact, the input to the filter seems to be wiggling about while the output is quiescent. WATSON becomes suspicious of the filter and expands it into the circuit diagram of figure 2.12. From *a priori* probabilities of failure he posits the shorting of C422. WATSON's model of this network (when operating correctly) is that C422 likes to keep node B from changing. That is, tugging on node A has no immediate effect on node B. Tugging on A for a while will eventually affect B. On the other hand, if C422 were to short, tugging on A could never affect B since the latter node would be at ground potential. This successfully explains the quiescent output of the filter. Is the behavior of the IF strip also explainable by this failure? Yes, since the behavior of the IF strip has already been explained in terms of the bad bias condition. Removing and checking C422, WATSON discovers that it does not meet its intrinsic specifications.

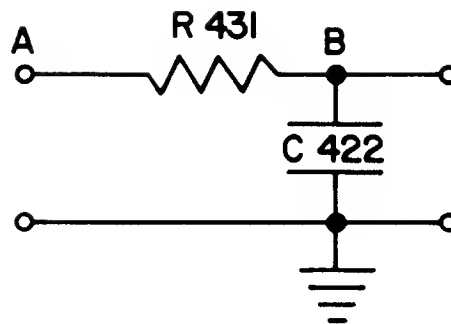


Figure 2.12 -- Circuit diagram of the low-pass filter.

Bug abstraction proceeds essentially as before. For the plan of figure 2.12 the symptom is a low input impedance for the filter network<sup>1</sup> and the sign is that the voltage across the capacitor is nearly zero. The mechanism for the bug is that the capacitor is behaving like a wire. For AGC2 (figure 2.11) the symptom is that the output is a lower bias voltage than expected. The sign is that the AVC amplifier has a lower gain than expected. The mechanism is that the low-pass filter loads the AVC amplifier so as to turn AGC2 into a source of 0 V. The symptom for the the RF section (figure 2.5) is that it has no discernible output, while the sign is a low bias voltage at the port between the IF strip and AGC2<sup>2</sup>. The mechanism is somewhat harder to construct in this case since it cannot be made directly from the parts of the plan for the RF section.

Recall that WATSON had to look inside the IF strip plan (figure 2.10) to decide what the IF strip would do if its control bias were not up to specification. WATSON deduced that especially strong signals would get through (though distorted), while weaker signals would be eliminated.

-----  
<sup>1</sup> Note that the output port of AGC2 (and consequently the output port of LO-PASS) is a voltage port, hence looking into a high impedance which should be reflected in a high input impedance for the filter network

<sup>2</sup> It is interesting to observe that a broken IF amplifier, due to an open transistor perhaps, would not exhibit the same sign as in the case of the broken AGC2. This is because the input to the IF amplifier from the AVC amplifier would be biased high, attempting to compensate for the broken IF.

The mechanism for the bug in the RF section therefore is that especially strong signals get through. Bias considerations inside the IF strip suggest that the bottoms of modulation signals may be clipped hence introducing distortion into those strong signals that are passed through. Since weak signals are expunged completely by the IF strip, inheritance implies their being expunged by the RF section as well. Abstraction of the bug to the top-level plan (figure 2.4) follows by composition of signal processing parts and inheritance of signal properties at ports shared at various levels in the plan-fragment hierarchy.

### 2.3 A misaligned front end.

In this scenario the presenting symptom of the GR-78 is that some station is audible in two places on the tuning dial. The plan of figure 2.4 tells WATSON that if the station is audible in one place, it should not be audible in the other. As before the signal generator is set up with a 5 MHz carrier and a 1 kHz modulation. The receiver is tuned to 5 MHz. The 1 kHz modulation is audible. The receiver's tuning control is then "swept" from 5 MHz downward. At a tuning of approximately 4.1 MHz the 1 kHz modulation is heard again without changing the carrier frequency of the signal generator. The plan of figure 2.4 indicates that the 5.0 MHz "station" should be inaudible. Successive refinement of the receiver's sweep setting shows that the output signal strength is maximized if the second tuning is set at 4.09 MHz.

With the experimental set-up described above still running, WATSON begins the back-trace. He finds that the RF section output (AF section input) is a strong 1 kHz signal -- which is wrong. Presumably the problem is not in the AF section since its input is a 1 kHz audio signal. Referring to the plan of figure 2.5, WATSON discovers a 1 kHz modulation component at the detector input and the IF strip input (converter output). The bad converter output is reasonable given that its input contains an RF carrier at 4.09 MHz<sup>1</sup>.

-----  
<sup>1</sup> This conclusion is obtained by expanding the converter plan and noting that the mixer is just as happy to mix down the *image* of the signal as it is to mix down the signal itself.

The trouble, then, seems to be in the RF amplifier. The teleological commentary on the RF amplifier indicates to WATSON that it serves two purposes -- to provide sensitivity and to provide selectivity. The first is achieved by amplification. Observation indicates that RF amplifier is successfully fulfilling this obligation. The second purpose includes providing a particular kind of selectivity called *image rejection*. This is achieved by passing a spectrum centered around the broadcast frequency with bandwidth significantly less than the IF frequency<sup>1</sup>. By observation the RF amplifier seems to be passing a significant image signal component. More explicitly it is failing to fulfill one of its design obligations. Hence LS3 suggests looking into the RF amplifier.

WATSON expands the RF amplifier into the plan of figure 2.13. He attempts to carry out a back-trace, but the measurements expert reports that back-tracing cannot produce reliable information since probing the inputs and outputs of the filters will result in changing their pass-band properties. WATSON must therefore pursue another failure hypothesis strategy. He notices that the two filters synergistically accomplish image rejection by successive "distillations" of the 4.09 MHz signal and successive "concentrations" of the 5 MHz signal. What relations must hold in order for this distillation process to work? The answer is that the two filters, which are narrow band-pass elements, must have the same center frequency (modulo stagger or offset, perhaps). (This chain of reasoning is really quite straight-forward, as the detailed account in sections 6.2 and 6.3 will reveal.)

In the present case the center frequency of each should be 5 MHz. Commentary requiring the filters to agree on center frequency suggests that *misalignment* as a plausible cause of the observed symptom. WATSON invokes the alignment expert to look into this problem, advising the expert that the two filters seem to disagree about the frequency of interest and that there are controls that are accessible to adjust the alignment of the filters. Note that the

-----  
<sup>1</sup> The bandwidth is actually much less than the IF frequency since the filters are known to be moderately high "Q" tuned circuits.

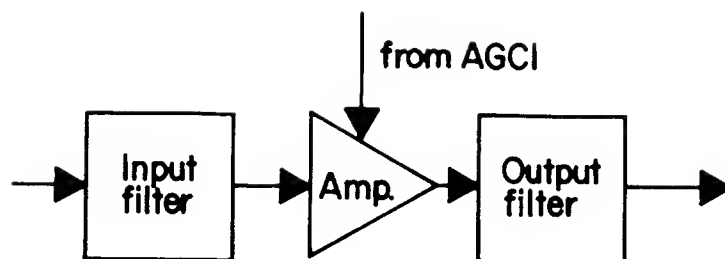


Figure 2.13 -- An expanded RF amplifier.

invocation of this expert was triggered by encountering a problem fitting the following general description: There is a sequence of band-pass parts whose purpose is to reject some frequencies and retain others. Their center frequencies' disagreeing would explain the observed symptoms. Moreover the overlap of those frequencies is adjustable by externally available controls. Note also that the alignment expert is non-local in the sense that it will probably have to appeal to plan knowledge unavailable in the plan within whose scope the expert was invoked. The loop breaking expert is non-local in this sense as well, since handling of loop structures requires simultaneous access to the details of the plans associated with the parts of the loop, and to the overall loop structure.

The alignment expert considers a "flattened" plan of the GR-78. This plan is obtained by manipulating the plan binding hierarchy that underlies the design, causing the innermost plans having band-pass commentary to appear at the top level. The alignment expert performs the usual task of first tweaking up the radio with respect to the high end of the band, then the low end. With respect to each end, the expert first sets the dial calibration by adjusting the oscillator (to yield maximal signal strength) and then proceeds along the signal path from antenna to detector adjusting each of the (adjustable) band-pass elements. This process is iterated until no improvement is achieved.

When the alignment expert finishes, WATSON looks to see if the problem has gone away.

Indeed there is no longer a significant 4.09 MHz component at the output of the RF amplifier. Bug abstraction has a new twist here in that it terminates at the level of the RF amplifier plan. When abstracting the bug at this level, instead of associating a sign with the symptom of strong image signal at the output, WATSON notes that the alignment expert should be invoked, followed by a verification that the problem has gone away. The bug mechanism is recorded as the enhancement of the image frequency with respect to the signal frequency. Bug abstraction at higher levels of the plan hierarchy goes through in much the same way as previously.

#### 2.4 An opened collector-base junction.

The symptomatology of this last scenario is that the radio seems to have no audio output. After some preliminary checking to verify that the radio has no audible output, independent of station selection and volume setting, WATSON calls for the usual set-up. Indeed, with the signal generator producing a 5 MHz carrier modulated at 1 kHz, the radio does not show a 1 kHz output. Referring to the plan of figure 2.4, WATSON determines that the inputs of the AF section meet specifications, whereas its output does not. Hypothesizing that the AF section is the culprit, WATSON pulls out its plan as represented in figure 2.14. The usual measurements indicate that the inputs to the audio amplifier too meet specifications, but the output does not.

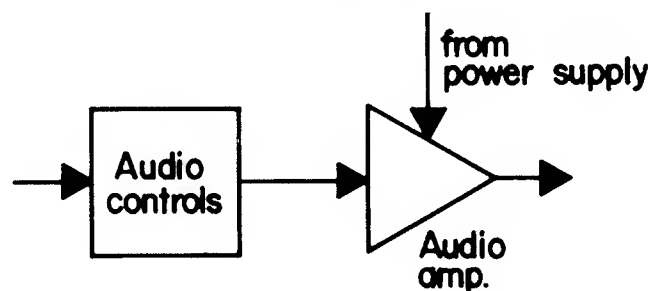


Figure 2.14 -- An expanded AF section.

Expanding the audio amplifier, WATSON finds the plan of figure 2.15. Tracing reveals



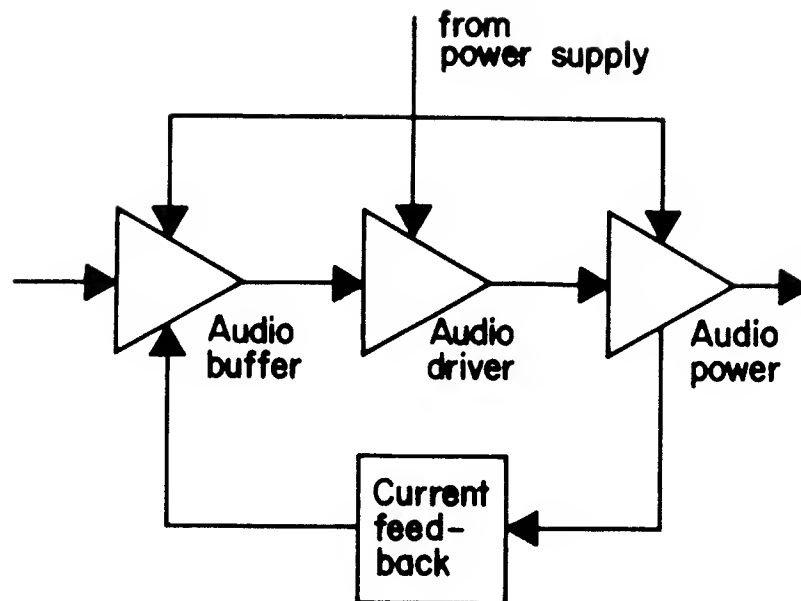


Figure 2.15 -- An expanded audio amplifier.

that the input to the audio power amplifier is quiescent -- which it ought not be<sup>1</sup>. A current probe placed at the input to the audio driver reveals a reasonable input signal<sup>2</sup>. WATSON therefore hypothesizes that the audio driver is the cause of the trouble and pulls out its plan. In order to understand WATSON's reasoning that will lead to localizing the failure, we will have to consider the circuit (as a whole) that underlies the audio amplifier plan of figure 2.15. There are of course plans for each of the parts in the block diagram of figure 2.15. The "dashed" boundaries encompassing the circuit components of figure 2.16 are approximations to those plans.

-----  
<sup>1</sup> The base biases of its transistors are also wrong in that they are pinned to ground, though the observation is not made at this level of detail. In signal back-tracing DC biases are typically ignored.

<sup>2</sup> The audio driver is current driven by its input signal, hence it is hard to measure that signal with great accuracy. In the present case a current probe will give sufficiently reliable information to determine whether or not the audio driver input is behaving reasonably, i.e. moving the voltage at the base of Q406 toward 9.6 V will cause the current into the base of Q407 to increase markedly. (See figure 2.16.)

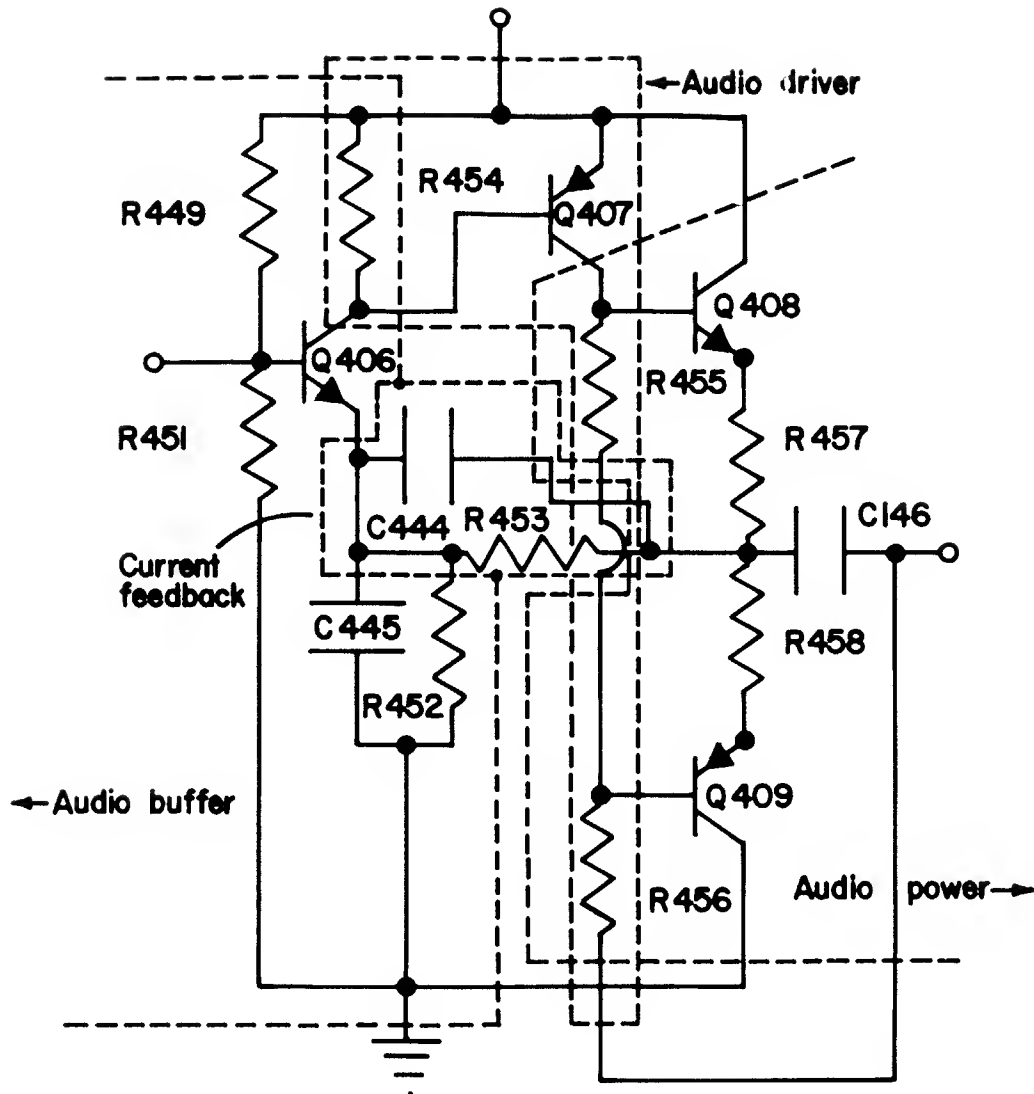


Figure 2.16 -- Circuit diagram underlying the audio amplifier.

WATSON wants to explain how the audio driver might have gone sour. Since he is looking at a circuit plan, he resorts to hypothesizing troubles in active components, because he

knows that their *a priori* probabilities of failing are greater than those for passive components. Consequently he focuses on Q407, a bipolar junction transistor (BJT). The most common way for such a component to fail is by the opening of one or the other of its junctions. From WATSON's point of view they (the opening of one or the other junction) are both equally good hypotheses, so he tries both, hoping to eliminate one or the other during the rationalization process.

This process goes forward in two phases. WATSON first predicts the DC consequences of the hypotheses and checks to see if the predictions correspond to reality. He then does a similar prediction and check for the AC consequences. Under the hypothesis of the open emitter/base junction the DC current in the collector/emitter branch of Q407 should become small. Now WATSON knows that Q407 biases its collector by injecting current into the series resistance composed of R455 and R456. If the collector/emitter branch current falls, the bias voltage at the collector of Q407 (base of Q408) falls as well. In the plan for the audio power amplifier R455 and R456 are used as a voltage divider that set the base bias for Q409. Since the voltage at the top of this divider falls under WATSON's present hypothesis, the voltage at the center falls as well. Thus the base bias of Q409 also falls toward ground.

The transistors Q408 and Q409 comprise a complementary symmetry pair. Under normal circumstances they are just barely turned on. If their base biases fall toward ground, they are completely shut off, though in falling, Q409 will at some point be turned on hard. In particular, neither transistor would have significant current in its collector/emitter branch. Hence the common node of R457 and R458 would fall toward ground too. This results in less current flowing through the feed-back resistor, R453. Therefore less current flows into the emitter node of Q406 from this network. Notice that Q406 is operating in its active region. In particular that means that this silicon BJT will produce as much current in its collector branch as is necessary to keep its emitter within 0.6 V of its base. Hence the current in that branch must increase to compensate for the loss of current formerly coming from the feed-back network.

Increased current in the collector/emitter branch of Q406 means more current coming

out of the collector node into the transistor. Symmetrically this means more current coming into the node from R454 and Q407. But WATSON has hypothesized the opening of the emitter/base junction! So the increased current must come entirely out of R454. This would result in a significant drop in the bias voltage of Q406's collector. Actual measurement indicates that this is not the case. This means that the hypothesis of the open emitter/base junction cannot be right. An alternative hypothesis, the opening of the base/collector junction of Q407, is tried. This leads to identical deductions up to and including the increased current in the collector/emitter branch of Q406. Again WATSON reasons that the increased current must come out of R454 and Q407. But now the emitter/base junction of Q407 is still a forward biased diode. Consequently the base of Q407, a germanium BJT, remains within 0.3 V of the emitter. So the bias at the base of Q407 remains fixed and the increased current actually comes out of Q407. All DC predictions are verified by measurement.

Now WATSON carries out the AC phase of the predictive process. The analytic method is not unlike that used to predict DC consequences. WATSON knows that a current encoded audio signal is presented at the input port of the audio driver. In order to understand the effect of such a signal on the audio driver he imagines the effect of positive and negative increments in current around the base bias current of Q407. He thinks of these increments as rising and falling at rates consonant with the 1 kHz audio signal that is present at the input port of the audio driver. To get the flavor of the analysis let's first look at the case of a normally operating Q407.

An incremental increase in current into the base of Q407 would result in an incremental increase in the voltage at the base of Q408 and an incremental increase in the voltage at the base of Q409. (Remember that R455 and R456 form a voltage divider.) This turns on Q408 a little, resulting in a larger (in magnitude) current flowing in its collector/emitter branch. This in turn means an incrementally larger current flowing to the right through the coupling capacitor C446. This has a positive feed-back effect on the voltage divider. That is, the base of Q408 is pulled

up even more. This is the mechanism by which Q408 is jolted into its active region. Symmetric reasoning on an incremental decrease in current into the base of Q407 leads to deducing an incremental increase in the current flowing toward the left through C446. This reasoning on voltages and currents lifts to the signal level as a voltage encoded signal at the output of the power amplifier that tracks the input to the audio driver. AC analysis must take care to note coupling capacitors like C446 and bypass capacitors like C445. Otherwise the local reasoning processes are indistinguishable from those applied in DC analysis.

Now to get back to the AC analysis of the hypothetical failure: An incremental increase in the current at the base of Q407 results in no change at the collector node. Similarly, an incremental decrease in the base current also results in no change at the collector. In terms of the signal at the output of the audio driver, this AC analysis would indicate no observable signal -- which is precisely the complaint. Pulling Q407 out and examining it shows that its collector/base junction has indeed opened, verifying the hypothesis.

In abstracting the bug for the audio driver, WATSON associates the symptom of a quiescent output in the face of an active input with a sign that includes the various bias changes that were predicted -- and observed -- in the audio driver. The mechanism follows directly from lifting the AC voltage/current behavior to the level of the abstract signals at the input and output of the audio driver amplifier. Abstractions at higher levels of planning are simpler compositions of signal processing.

### 3 A Preview of the WATSON Program

Before delving into the details of the machinery enabling WATSON to carry out his task, we should stand back and look at his overall structure, developing a picture both of the major parts of his anatomy, and of their interactions. Figure 3.1 shows a flow diagram for WATSON. In that time-worn tradition, polygons denote essentially procedural structures, and balloons denote bodies of knowledge. Control flows along the solid arrows, while information flows along the dotted arrows. This diagram is somewhat fictitious, first because it is an incomplete representation of WATSON's control structure. Second, the segregations suggested by the closed figures are conceptual and do not necessarily correspond to independent realizations within WATSON. Nevertheless, the flowchart presents many of the essentials of the CONNIVER and LISP functions forming WATSON's control structure, together with the CONNIVER data base in which his knowledge resides.

#### 3.1 Stating the problem.

WATSON accepts complaints about some particular radio design. A complaint is formulated as a pairing of inputs and outputs. The intended interpretation of this pair is that the outputs are incorrect given the inputs. A complaint is formalized as an s-expression (read by WATSON) of the form<sup>1</sup>

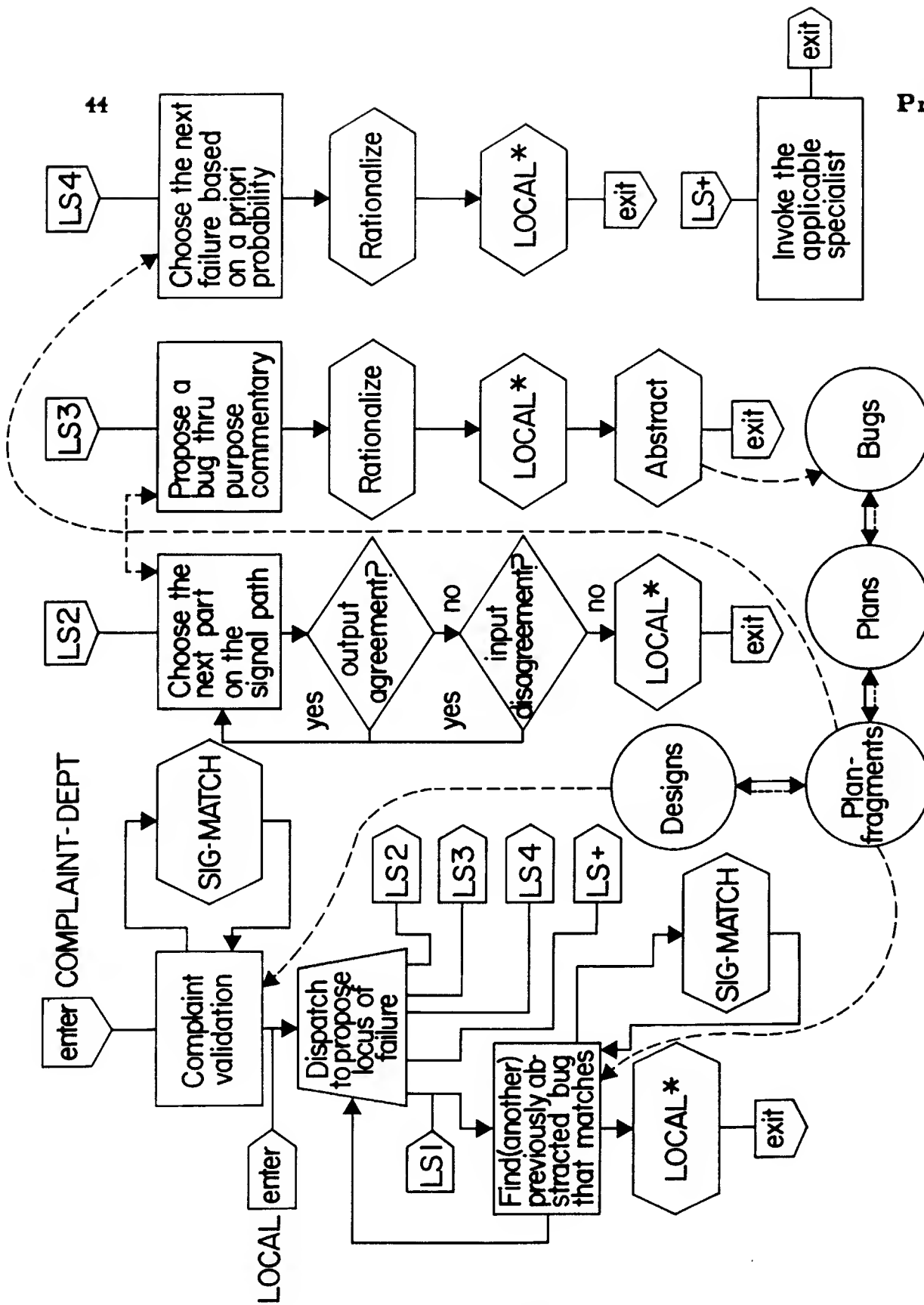
```
(COMPLAINT
  design
  list-of-input-signals
  -----
```

<sup>1</sup> Syntactic variables will be indicated by the use of the lower-case Roman font. Optional structures will be enclosed in "chevron" angle brackets. Alternatives will be indicated by vertical bars.

list-of-output-signals  
«control-variable-bindings»).

The COMPLAINT may variabilize various properties of the input and output signals, declaring the kinds of values these properties may take on. WATSON first asks himself if the COMPLAINT is reasonable. The determination of reasonableness is made by first matching the inputs mentioned against whatever expectations for such input signals might be specified in the design. If the inputs fail to match such specifications, the COMPLAINT is ill-founded, and WATSON so informs the plaintiff of his error. Otherwise WATSON proceeds to determine whether or not the mentioned outputs jibe with the mentioned inputs. This latter determination is made by first inferring the expected outputs of the radio receiver from the reported inputs by the use of various rules. The inferred outputs are then matched against the outputs reported in the COMPLAINT. If the match succeeds, the COMPLAINT is ill-founded. Otherwise, the COMPLAINT is valid, and WATSON retains a record of the mis-match so as to facilitate the construction of a test bench set-up (for debugging purposes).

COMPLAINT validation is somewhat complicated by the fact that a COMPLAINT need not completely specify the operating configuration of the faulty receiver. Thus external controls, for example, may provide a number of degrees of freedom in operating specifications which must be pinned down when debugging. In fixing these degrees of freedom (which could have been specified completely in the COMPLAINT) WATSON may conclude that the COMPLAINT is ill-founded because of unreasonable configuring of the receiver's controls. For example, a COMPLAINT suggesting that a receiver has no audible output, but not specifying the volume control setting, may be determined to be ill-founded on the grounds that when the volume control is set at a reasonable level, the receiver does have an audible output. The operating configuration for the debugging procedure is actually determined in the same operations as validation of the COMPLAINT with respect to the settings of external controls. The fixing of the operating configuration is a consequence of the process of establishing the validity of the COMPLAINT under whatever constraints the COMPLAINT imposes.



Preview

Figure 3.1 – Failure localization in WATSON.



### 45 Localizing the failure.

When a COMPLAINT is lodged, WATSON is entered (see figure 3.1) at the label, COMPLAINT-DEPT, and is validated as per the discussion above. Then the real work begins with a call to LOCAL. As mentioned previously, WATSON has available a number of localization strategies, but as with all reasonable problem solvers, he always attempts to use the strategy which asks, "Is the answer already known?" Hence, he first dispatches to LSI to see if a bug has been previously abstracted that covers the present situation. The principal determiner of the applicability of an abstracted bug is the general signal matching processor, SIG-MATCH, but as we shall eventually see, other recognition criteria may be in order as well. If a bug fits, WATSON is left with a sub-plan-fragment (of the plan-fragment to which LOCAL is currently being applied) which contains the locus of failure.

The dispatch mechanism is not a simple COND, of course, for the strategies dispatched to may be exited at various stages of completion. LSI may run out of applicable bugs, for example. Or the successful localization at one level may not yield a successful localization at the next level. Hence the hypothesis implied by a localization (that a particular sub-plan-fragment is at fault) would have to be suspended to admit trying some other possibility. The dispatch is further complicated by the fact, for example, that the back-tracing strategy, LS2, may make immediate reports to the current activation of LOCAL about interesting phenomena it (LS2) encountered. Such reports may cause the suspension of the back-trace and the activation of some other strategy.

Note that LSI relies on the body of knowledge in the 'plan-fragments' balloon. That is not to say, however, that that is the only source of information. The double-headed solid/dotted arrows indicate that information from one body of knowledge may be fetched (or added) through another such body. The mechanisms for doing these indirect acquisitions of information are typically embedded as CONNIVER methods in the data base.

If no applicable bugs are known, WATSON looks around for another localization strategy, one of which is LS2, the back-tracing strategy. This strategy is applicable to cascade plans, and is

an iteration on parts found along the signal path of such a plan. The basic iterative step is to examine whether or not the observed output of a part agrees with the predicted output (given the parts observed inputs). If a disagreement is detected, the strategy next looks to see if the expected and observed inputs agree. If they do, the part "sandwiched" by those inputs and outputs is presumed to be at fault. The polygon labelled 'LOCAL' suggests a recursive call (denoted by the star) to LOCAL. LOCAL is applied to the plan-fragment corresponding to the "found" part. Once again, I should like to emphasize that the flowchart omits a great deal about the flow of control in WATSON. Several kinds of failure are possible on the LS2 branch of the localization process. Such failures can cause a number of different changes in the flow of control, ranging from the temporary suspension of a hypothesis (of failure of a part along the trace path), to complete abandonment of the LS2 branch. LS2 draws principally on knowledge from the 'plan-fragments' balloon.

The LS3 strategy is typically entered by virtue of LOCAL's having received a report from some other strategic branch indicating that some well-defined difficulty has been encountered. "Well-defined-ness" is determined by the failure to meet some criterion of the design. This suggests checking to see where the responsibility lies for meeting such a criterion. If a sub-plan-fragment with the appropriate purpose commentary can be located, an investigation ensues to see if the failure to meet the responsibility (by the identified part) could lead to the observed misbehavior in the local plan-fragment. If this rationalization is successful, a process of abstracting a description of the bug and its underlying cause ensues, and a record of the abstraction is made in the data base under the heading 'bug' as a note on the plan of which the local plan-fragment is an instance.

LS4, the strategy based on *a priori* probabilities of failure, is only applied when WATSON has come within immediate reach of resolving the underlying cause of the initiating COMPLAINT. This is the case when the problem has been localized to a plan of the circuit or coupling type. At that point sub-plan-fragments will typically be instantiations of plans of the component type.

hence will be annotated with descriptions of their failure modes, including likelihood of occurrence. The failures are sorted in the order implicit in the annotation, are selected from in that order, and are rationalized with respect to the observed AC and DC behavior of the circuit (coupling) plan. A successful rationalization will lead to a recursive invocation of LOCAL. This invocation will terminate the localization process if it is applied to a plan-fragment corresponding to a component, resulting in the pulling of that component for inspection of its intrinsic properties.

Finally there is the LS+ branch of possible localization strategies. This is the catch-all strategy which invokes various specialist experts based on the encountering of particular kinds of structural impediments to the other localization strategies. There is presently only one such expert contemplated for implementation -- the loop breaking expert. Others, however, are easily imaginable. For example, it might be appropriate to have an expert to deal with 60 Hz hum, a common enough bug in radios. When this phenomenon is encountered, there is usually no particular source identifiable for it. In any event experts appropriate to the local structure of the localization problem will in turn call upon other control and knowledge structures in WATSON, including LOCAL.

### 3.3 Describing radios to WATSON.

Besides the failure localization subsystem, WATSON's other major subsystem is that concerned with plan description and design assembly -- which we shall call the 'assembly subsystem'. The interesting aspects of the control structure of this subsystem would not be particularly elucidated by a flowchart, hence I have chosen not to present one. The major functions of the assembly subsystem include the definition of plans for various functional units in a radio receiver, the compilation of instances of plans as plan-fragments, and the association of plan-fragments in the hierarchy of a radio design. The first function is carried out in a very

straight-forward fashion. The second function -- compilation -- introduces complexity due to the fact that much of the essence of a plan-fragment is procedural in nature. These procedures are stated in a stylized form as rules associated with a plan. The application of a rule may be triggered by various combinations of events (items ADDED to the CONNIVER data base). The particularization of a rule to a plan-fragment turns out to be a rather complex process involving the creation of many CONNIVER functional closures.

The integration of plan-fragments into a design is complicated by the fact that design is typically carried out in a "top-down" fashion that delays the filling in of various "slots" specified in a plan. The delay is due to the fact that the slots can be filled explicitly by subsequent introduction of design sub-structure in the assembly process. Similarly the filling in process may happen implicitly at run-time. By 'run-time' I mean the point at which the design is used to do reasoning about the expected behavior of the radio receiver. As it turns out, all such delayed references are resolved using CONNIVER IF-ADDED and IF-NEEDED methods. Needless to say, the result of entering the assembly subsystem is the adding of facts to the data base under the balloon headings 'plans', 'plan-fragments', and 'designs'.

### 3.4 Programming constructs.

In the succeeding chapters I shall frequently exhibit fragments of CONNIVER or LISP code. I shall take the liberty of assuming familiarity with the primitives of those languages, as they are well documented elsewhere [McCarthy, 1965; McDermott, 1974b; Moon 1974]. Functions that are peculiar to WATSON will be mentioned as such and will have their semantics explained (if not explicitly, then by context). Most such code fragments will make reference to the CONNIVER data base. Whenever possible, I have tried to avoid the cumbersome CONNIVER pattern matching syntax by using the lower-case Gothic font for pattern-variable names. The nature of the intended match should be clear from the textual context.

There are three syntactic constructs introduced and used extensively: FINDs, FORs, and *path names*. A FIND has the general form

(FIND quantifier variable-list pattern),

where 'quantifier' is a quantifying expression over the list of variables, 'variable-list'. The meaning of the quantifier will always be self-evident. The basic idea is that the mentioned pattern (possibly containing match variables bound in the scope of the FIND, possibly containing match variables free with respect to the FIND), should be matched against in the data base under the constraints implied by the quantifier. For each successful retrieval of a matching data base item, the outstanding bindings of the bound variables are appended to a list. The list is returned as the result of the FIND after all possible matches are made. It will be my habit to mention variables bound in the scope of the FIND as if their bindings were still available on exiting the FIND. I shall do so however only when context makes my intentions unambiguous.

A FOR, which has the form

(FOR quantifier variable-list pattern prog-body),

is much like a FIND. Instead of returning the bindings resulting from successful matches with the data base, for each successful match, the sequence of code 'prog-body' is evaluated. Of course, the code may mention the variables bound by the FOR, which will have values appropriate to the most recent match. I should mention that patterns in both FINDs and FORs will usually be QUOTEd (since that argument position is passed by value). Typically, however, instead of the usual LISP quotation, I shall use the CONNIVER skeleton construct '! '.

Finally, though all the objects which the assertions in the data base are about generally have canonical names, such names are not exactly rife with semantic content. For clarity I shall generally make use of the path-name construct

~(NAIL INDEX-FINGER RIGHT-HAND G00069),

which denotes the obvious part of a distinguished individual, G00069. This particular path-

name evaluates to the intended canonical object at read-time. There are also other path-name macro characters -- `'`, `~`' and `'`, `,`, `~`' -- which cause path-name evaluation at EVAL time and pattern match time respectively.

## 4 The Case of Q301

Having seen what WATSON can do, we are now in a position to look into how he does it. The rest of this exposition will be devoted to explaining the "how." We will accept WATSON's initial knowledge as *fait accompli*, ignoring for the time being the intricacies of the readers and evaluators that make it possible to communicate to WATSON the large data base that encompasses plans, designs, component descriptions, circuit diagrams, etc. We will, however, be much concerned with the internal representations that these data assume, and the manipulations that are applied to them.

Another aspect of the communication problem that we will ignore here is the digestion of the sensory data provided by measuring instruments. These data may be as trivial as a reading from a voltmeter, or as complex as a signal trace on an oscilloscope face. Neither of these is the real object of the measurement activity. The real object is the association of values with the obs of an abstract signal description. Ideally WATSON would be able to take these basic data and translate them. He does not in fact do this. WATSON's measurements specialist, MAXWELL, to which I shall make frequent reference, presents the human assistant with questions phrased in terms of the obs needing value assignments. Hence the burden of translation from sensory data to symbolic description is carried by the human assistant.

### 4.1 Lodging the complaint.

Recall that in the first scenario (section 2.1) that the presenting symptom was that the GR-78 showed no output. The formal statement of this complaint is

(COMPLAINT GR-78

```

(INPUTS
  (~(OBS-SIGNAL PORT-1)
    (CARRIER-COMPONENTS
      (MODULATION AM)
      (CARRIER-FREQ x)
      (MODULATION-COMPONENTS
        ((MODULATION-FREQ y ∈ {*}) ) ) ) ) ) )
(OUTPUTS
  (~(OBS-SIGNAL PORT-2)
    (CARRIER-COMPONENTS
      (MODULATION AM)
      (CARRIER-FREQ 0.)
      (MODULATION-COMPONENTS
        ((MODULATION-FREQ z ∈ {*})
          (AMPLITUDE 0.)) ) ) ) ) )
(CONTROL-BINDINGS
  (TUNING x ∈ {*}) ) ).

```

A COMPLAINT is lodged about a particular design, the GR-78. Unhappily, the syntax of COMPLAINTs is quite complicated<sup>1</sup>, but the gist of it is that we are displeased with the observed output at the port named PORT-2<sup>2</sup> given the observed input at port PORT-1, and given that the GR-78's external controls are set up in a certain way. WATSON's first order of business is to validate the complaint experimentally and to use the experimental results to design a test set-up for tracking down the bug. To do an experiment, settings of the signal generator controls and the radio must be chosen. The "declaration"

...CARRIER-FREQ x...

tells WATSON to name the carrier frequency parameter of the generated signal 'x', but offers no advice as to how it is to be assigned. The CONTROL-BINDINGS clause specifies which of the radio's control variables -- TUNING (i.e. radio's station selection control) -- is to be affiliated with the carrier of the input signal, x, and specifies that it can be assigned any legal value. WATSON

-----

<sup>1</sup> WATSON thinks of an audio signal as a modulated DC signal.

<sup>2</sup> Many of the abstract objects of WATSON's knowledge base -- ports, nodes, plan-fragments, etc. -- are realized by atomic objects called obs with GENSYM canonical names of the form FROB-105. Such obs can generally be gotten at in either of two ways, by the structural route -- as exemplified by the kinds of data base items we shall be looking at presently -- or by path name. The "gritch" character, '~', is a reader macro character indicating that the subsequent list-structure denotes a particular ob.



finds the range of possible values for that control variable. He can do that because the data base contains the items<sup>1</sup>

```
(PLAN RECEIVER CASCADE)
(PF PF-3 RECEIVER GR-78)
(PF-PART NIL NIL PF-3)
(PF-CTL PF-3 TUNING CTL-4)
(VALUE-RANGE CTL-4
 (CONTINUOUS-GENERATOR (FROM 3000000.) (TO 7500000.)))
(OPTIMAL-VALUE CTL-4 5000000.).
```

Now knowing that the value range is continuous between 3 MHz and 7.5 MHz, WATSON decides to do experiments for 3 MHz, 5 MHz, and 7.5 MHz settings of the radio's station selector.

The input signal described in the COMPLAINT also has the following declaration

...y  $\in$  {\*}...

which says that the value of MODULATION-FREQ should be set to whatever is convenient for making the generated signal match the expected input signal for the GR-78. Not unlike the optimal value (the frequency at which the plan works best) seen above, the expected input signal has an optimal value which is identified with the audio mid-band frequency of the radio's AF section, 1 kHz. There is also commentary indicating that the modulation frequency can vary continuously from 100 Hz to 5 kHz. WATSON therefore chooses experimental settings for the modulation frequency of 100 Hz, 1 kHz, and 5 kHz respectively. z, like y can be set to anything that is convenient. Since it is an output parameter, however, 'anything' means that it can potentially take on all possible values. That is, every possible modulation component on the output has negligible amplitude<sup>2</sup>. This exhausts the degrees of freedom mentioned in the input signal.

-----  
<sup>1</sup> Some of these items are not actually PRESENT in the data base in the formal CONNIVER sense. They may appear by virtue of various deductive methods. It will suit our purposes for the time-being to imagine that they are PRESENT.

<sup>2</sup> This follows from WATSON's essentially linear model of the signal processing carried out by radios.

The next order of business is to use those degrees of freedom implicit in the controls of the radio. Evaluating

```
(FIND ALL (x) ' (CONTROL (T-L-PLAN-FOR GR-78) x))
```

yields a list

```
((PF-CTL PF-3 TUNING CTL-4)
 (PF-CTL PF-3 VOLUME CTL-5))
```

of possible bindings for the match variable, *x*. The first item has already been taken care of by virtue of having generated the three possible station settings. The second item is completely free. A query for the value range of CTL-5 reveals to WATSON that VOLUME is continuously variable over some range, a pair of numbers corresponding to the "full-scale off" and "full-scale on" positions of the volume control knob. He chooses the end-points and middle value for the purposes of the validation experiments.

The result of all this is that WATSON has generated a number of possible configurations for the signal generator and radio by the obvious combinatorics. Actually there are two degrees of freedom, that have not been considered in configuring the experiments. It so happens that the generator's amplitude and percent modulation are selectable. It is WATSON's heuristic inclination to leave these at *a priori* fixed values unless these signal properties (amplitude and percent modulation) are mentioned specifically in the COMPLAINT. Also, the GR-78 design has an input port not mentioned in the COMPLAINT, the power port. WATSON knows about three kinds of ports, those with control signals, those with information signals, and those with power (or bias) signals. WATSON proceeds to validate the complaint in the scope of a loop whose rounds are determined by the configuration combinations developed above. (There are of course, twenty-seven such inner-most rounds.) Forward reasoning is required to deduce what the output should look like given the inputs established by a particular test configuration. The forward reasoning is actually carried out in a CONNIVER context<sup>1</sup> that is private to the current inner-most round. Figure 4.1

<sup>1</sup> Since I shall have the occasion to use the word 'context' in a number of ways, 'context' in the lower-case Gothic font is reserved for the CONNIVER usage of the word.

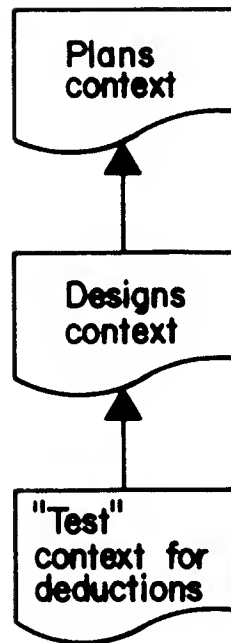


Figure 4.1 -- The context structure at the time of validation.

illustrates WATSON's use of the context machinery. At the top-level is a context containing the abstract plans along with various facts. Below that is a context containing the GR-78 design and whatever facts are local to that design. Finally there is the context in which the deduction will happen.

How is this forward reasoning done? Answering this requires some explanation of the theory of representation embodied in WATSON. We should first realize that most of the items in his data base are propositions over a class of objects called *obs*. *obs* are carriers<sup>1</sup> for the attributes represented by the propositions, the latter being realized as CONNIVER items. Earlier in this section we saw the following propositions

(PLAN RECEIVER CASCADE)

<sup>1</sup> It may also be helpful to think of them as analogous to the formal objects that Sussman [1975] uses in subroutine generalization, or to the anonymous identifiers used by Hewitt [1971] to carry out procedural abstraction, or to Skolem functions [Chang, 1973] in their standard role in the elimination of existential quantifiers.

(PF PORT-2 RECEIVER GR-78)  
 (PF-PART NIL NIL PORT-2)

which essentially said that there is a plan- fragment whose unique ob name is PORT-2 and it is a token (at the top-level) of a plan called RECEIVER. PORT-2 is the ob of interest; it is the instantiation of the RECEIVER plan as the top-level plan of the GR-78. One particularly interesting proposition satisfiable by an ob is the VALUE proposition. Specifically,

(VALUE PF-3 3500000.)

corresponds to the TUNING control indicating that the GR-78 is tuned to 3.5 MHz. The VALUE-RANGE proposition, which we have also seen, may also be satisfied by an ob.

In the validation situation, what WATSON really wants to know is whether or not the signal reported<sup>1</sup> at the speaker port of the GR-78 is reasonable, given the signal reported at the antenna port. Now the input signals generated in each of the validation test cases that he creates are instances of the reported input signal. Let SIG-6 and SIG-7 be the ob names denoting the generated signal and expected signal structures, respectively. Evaluating

(SIG-MATCH 'SIG-7 'SIG-6)

will tell WATSON whether or not the signals match<sup>2</sup>. If they do, the input is declared to be reasonable, given the design. A successful match leaves the values of the various obs bound, or at least "ranged," in the context of the current experiment.

WATSON matches the reported output signal against the expected output signal. However, he will find that many of the obs in the output signal structure, SIG-8, do not have

-----  
<sup>1</sup> WATSON distinguishes signals reported in a COMPLAINT from those that he observes.

<sup>2</sup> We need not worry yet about the precise description of the signal structures denoted by SIG-7 and SIG-6, nor about the precise nature of the matching procedure SIG-MATCH. For the time being it should be adequate to think of this latter procedure wandering (in parallel) over a pair analogous structures of a particular kind. The matcher embodies a theory of compatibility for various pairings of ob values and ranges. The procedure reports whether or not visited pairs of obs satisfy this theory. Whenever a matching pair is found, a VALUE-RANGE for that pair is created in the context of the match. A mis-match occurs whenever a VALUE-RANGE is forced to be empty. Hence the matching operation is a test for non-empty intersection.

value properties. What they do have is commentary indicating how such values might be deduced from values of obs of the input signal description, SIG-7. Briefly the value is deduced through the interaction of a number of CONNIVER methods. The first kind of method triggered is an IF-NEEDED -- called a determiner -- whose pattern of invocation matches

```
(VALUE REG-9 value),
```

where REG-9 is an ob whose value is needed for the match of SIG-8 with the observed output signal. The determiner in turn fetches and asserts a number of items. These assertions have the effect of triggering IF-ADDEDs that are the constituents of rules. These latter methods ADD items to the data base that look like

```
(RESULT result-name
  (rule . list-of-results)
  (VALUE REG-9 value)).
```

This says that a kind of fact called a RESULT, whose name is 'result-name' has been derived by the named 'rule' from other facts on 'list-of-results'. The fact in question concerns the value of REG-9. Rules are built up from a complex of IF-ADDED methods whose presence in a context may depend on what RESULTS have already been ADDED to the data base. Thus in evaluating REG-9, the values of its antecedents are asserted (as RESULTS) which will then trigger a rule producing a RESULT giving the value of REG-9. This value finding is recursive in that values for the antecedent obs may have to be gotten by the obvious reinvocation of the same machinery<sup>1</sup>. The io-contour mentioned in section 1.2 is the closure of an ob under RESULTS that determined its properties together with RESULTS it helps determine.

Now WATSON wants the output match to fail. (Remember the COMPLAINT claims a discrepancy between the observed output and the expected output.) The match does fail. Why? Suppose that obs REG-10 and REG-11 correspond respectively to the modulation frequency and

-----  
<sup>1</sup> In the present case the only way a ob of the input signal description can fail to have a value is by virtue of not having told it to WATSON. Consequently such missing values are found by appealing to the measurements expert.

modulation amplitude of the expected output signal. Let REG-12 and REG-13 be the analogous obs in the observed output signal. Finally let REG-14 be the modulation frequency of the antenna input signal. Via the determiner query above, REG-10, REG-12 and REG-14 are discovered to be equal, as they should be. REG-13 unfortunately has a value of 0, belying the non-zero predicted value associated with REG-11. No match! In fact, any of the twenty-seven experiments in which the volume control is not specified to be at its low end leads to this mis-match. WATSON's final step in each round of the validation loop is to make sure that the measured (observed) signal matches the reported signal. This turns out to be the case, meaning that the COMPLAINT is valid.

Now to establish a test set-up for the localization of the failure, WATSON recollects how the preceding experiments were generated. For all of the continuously ranging control variables (corresponding to certain degrees of freedom in the experiments) he selected high, middle, and low values. High and low values can be gleaned from the salient VALUE-RANGE propositions. The middle value is typically an optimal value (for the performance of the radio), also gotten from explicit commentary, or a mean of the high and the low values. The test set-up will typically be derived from these middling values. In particular, for the rest of the first scenario, the signal generator and receiver are both tuned to 5 MHz with a 1 kHz modulation being imposed on the signal generated. The volume control is at its medium value, and the amplitude and percent modulation of the input signal remain at their "typical" values.

#### 4.2 Signal tracing.

Having satisfied himself that there really is a problem, and having devised a test case which reliably demonstrates the problem, WATSON invokes his localization specialist, LOCAL. As we shall shortly see, this is a recursive procedure, whose recursive structure parallels the hierarchical structure of plan-fragments embodied in the design of the radio receiver. The first question asked in the localizer is whether there are any known explanations for the observed symptomatology. Evaluating

```
(FIND ALL (bugname sign mp)
  (BUG bugname RECEIVER
    (SYMPTOMS PF-3
      (INPUTS (PORT-1 SIG-6))
      (OUTPUTS (PORT-2 SIG-8)))
    (SIGNS . sign)
    (MISSING-PURPOSES . mp)))
```

will return all previously abstracted bug descriptions whose symptomatologies match the present case. (Remember that PF-3 is the plan-fragment instantiating the top-level plan for the GR-78.) In the scenarios we have seen, WATSON is presumed to have done no previous debugging, nor is he initialized having abstract bug knowledge associated with plans. Consequently this query in the localizer is unenlightening for WATSON.

Having no *a priori* knowledge about the particular problem facing him, he begins to trace the signal. Understanding this process requires understanding more of the details of the representation of plans and plan-fragments. We have already seen the items

```
(PLAN RECEIVER CASCADE)
(PF PF-3 RECEIVER GR-78)
(PF-PART NIL NIL PF-3)
```

that tell WATSON that PF-3 is an instance of the RECEIVER plan, at the top-level of the GR-78 design. WATSON also knows that this plan is of type CASCADE, hence it is a reasonable candidate for signal tracing. To do the tracing, however, he needs to know the port/part interconnection description provided by the RECEIVER plan. Items of the form

```
(PORT-SANDWICH RECEIVER part-name1 port-name part-name2|NIL)
(PF-PORT PF-3 port-name can-name)
(PF-PART PF-3 part-name plan-fragment)
```

are sufficient to give this information and to map abstract parts and ports of the plan onto realizations in the design. WATSON already has his hands on the canonical name of the speaker port of the GR-78 -- PORT-2. Three queries of the data base will yield both the parts of the RECEIVER plan that are looking into the speaker port, as well as the names of their plan-fragment instantiations in PF-3.

It is helpful to think of PF-3 as being "active" at the current level of WATSON's recursive localization process. We have already seen that the ports of PF-3 have structures hanging on them that represent the signals at those ports. Such structures are built up from data base items of the form

```
(PF-SIGNAL PF-3 PORT-1 SIG-15)
(MODULATION SIG-15 REG-16)
(VALUE REG-16 AM)
(CARRIER-COMPONENT SIG-15 REG-17)
```

which tell us some of the features of a signal (under a frequency domain interpretation) at the GR-78's antenna port, PORT-1<sup>1</sup>. What I should like to emphasize is the fact that such signal descriptions are local to some plan-fragment -- PF-3 in this case. Why aren't signals globally known as are the ports themselves? Observe that the plan-fragment PF-3 is an instance of the RECEIVER plan, and as such must have sub-structures corresponding to the plan's parts. These parts are associated with other plan-fragments that are instances of yet more plans. Consider then that the hierarchy of plan-fragments, that this organization entails, represents various levels of detail in the design. As WATSON delves deeper into the hierarchy by successively activating various plans, more features of a signal at a given port should become relevant. If plan-fragment A encompasses plan-fragment B (as a part) in such a way that they share a port, there are details of the signal at that port which are appropriate for B to know about, but not A. This problem of local visibility is solved by associating two copies of the signal at the port, one for A and one for B. As we have already seen, part of the strategy for evaluating the obs that comprise the signal is to look for a value on the local copy of the signal, or by determination from its antecedents. If none is forthcoming, the ob evaluator looks up the hierarchy of plan-fragments to see if the value can be found on an analogous ob associated with the signal at the same port. One more

-----  
<sup>1</sup> Note that the value of the modulation ob is available directly. Again let me emphasize that we will generally assume immediate availability for the sake of simplicity, though such values may actually be found by a determiner or other deductive methods, such as inheritance through the plan-fragment hierarchy.



complication in the representation of signals is the fact that the signal actually appearing at a port may bear no resemblance to the signal that is expected there. The signals that I have been referring to are the expected variety. In parallel with each occurrence of a structure representing the expected signal, there is also a structure representing the real, or observed signal, as is indicated by

(PF-OBS-SIGNAL PF-3 PORT-1 SIG-18).

Now we know enough about the representational machinery to describe the basic signal-tracing localization strategy in some detail. We presume that the strategy is at some arbitrary level in its recursive invocation and have focused on the output port P in plan-fragment PF as having an offensive signal. PF is declared to be the active plan-fragment by binding the CONNIVER variable CULPRIT to PF. Another aspect of activation is the pushing of the old context and the creation of a new one. This corresponds to the fact that WATSON is hypothesizing PF as the source of of the difficulty, a hypothesis that WATSON may wish to back out of at a later time. Such backing out is made very convenient by contexts, since they admit quick dismissal of deductions predicated on the hypothesis that generated the context.

Suppose PF is an instance of the plan illustrated in figure 4.2. WATSON walks along the cascade, internally represented by a collection of PORT-SANDWICH propositions. Beginning at the output port of the rightmost part, WATSON does a SIG-MATCH of the expected signal with the observed signal. Matching against the expected signal may, of course, necessitate propagating (by determiners and rules) the leftmost input signal through the cascade formed by the various plan parts. This process goes as previously described, with the required obs being filled in with their values. The general strategy is to move leftward along the signal path until a part is reached at which the SIG-MATCH succeeds on the output side but fails on the input side. Note that PF must have taken the blame in just this sort of process. Since some part internal to PF shares its good input port, the leftward movement must terminate. This strategy also works in the more complicated case of figure 4.3. The occurrence of a part join requires the tracing process to split

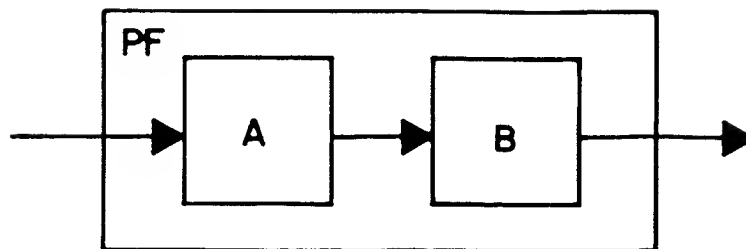


Figure 4.2 -- A simple cascade.

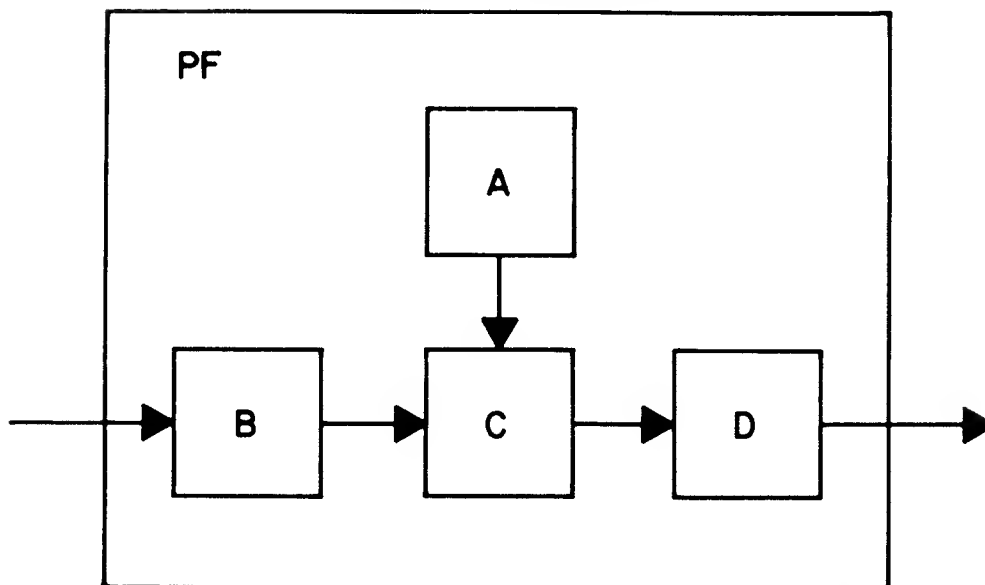


Figure 4.3 -- A cascade with a simple part join.

into two subprocesses. It should now be realized that the set of SIG-MATCHes performed is somewhat more complicated than I actually stated. For each input port of a part, the observed signal is SIG-MATCHed against the expected signal. All such matches must be successful in order to terminate the trace propagation. The trace only propagates at ports showing mis-matches.

Figure 4.4 shows the most general case of tracing topology. The existence of loops in the signal necessitates that the tracing process mark the parts as they are visited. But since there

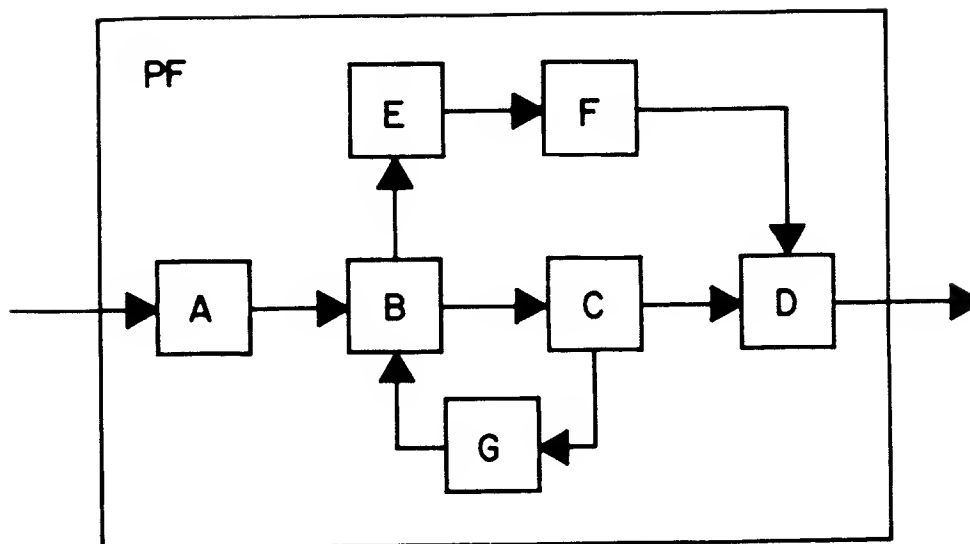


Figure 4.4 -- A fork and join combined to give a loop.

are potentially many instances of the tracing process making visits (consider the fork in the figure) a marking scheme must indicate who made the visit. Things are further complicated by the desire to merge tracing processes whenever possible. Note that only one tracing process need emerge through the input port of B. An ancestral process naming scheme, combined with a "visiting card," indicating the port used by the uniquely identified process in making its visit, solves all of these problems.

A straightforward application of the signal-tracing localization strategy, LS2, underlies the first scenario. The initial recursive application is to PF-3, an instance of the RECEIVER plan. Items in the data base:

(PF-PART PF-3 RF-SECT PF-19)  
 (PF-PART PF-3 AF-SECT PF-20)

indicate that this plan has two cascaded parts. Tracing the cascade reveals that the input signal to PF-21, the realization of the AF section in PF-3, does not meet specifications. A partial comparison -- component by component -- of the observed and expected signals

```

(PF-OBS-SIGNAL PF-20 PORT-22 SIG-23)
(PF-SIGNAL      PF-20 PORT-24 SIG-25)
(CARRIER-COMPONENT SIG-23 REG-26)
(CARRIER-COMPONENT SIG-25 REG-27)
(MODULATION-COMPONENT REG-26 REG-28)
(MODULATION-COMPONENT REG-27 REG-29)
(MODULATION-AMPL REG-28 REG-30)
(MODULATION-AMPL REG-29 REG-31)
(VALUE REG-30 0.)
(VALUE REG-31 5.)

```

shows that there are different modulation amplitudes, causing SIG-MATCH to fail. Since it turns out that the inputs to PF-19, the plan-fragment corresponding to the RF section, meet specifications, whereas its outputs do not, PF-19 becomes the CULPRIT. The localization process then enters a new level of recursion.

#### 4.3 The loop problem solved.

This recursive localization process continues smoothly until arriving at PF-32, an instance of the plan, OSC. Figure 4.5 shows the various recursive activations of LOCAL. PF-32 got the blame because SIG-MATCH decided that the observed output of the oscillator did not match the expected output. (Note that the oscillator has no inputs apart from power.) In particular, the output is a quiescent DC bias. LOCAL tries the usual trick of expanding the OSC plan into its parts, finding the corresponding plan-fragments, and examining the i/o properties of those fragments. In the present situation, the tracing process finds itself in a loop. (Refer to figure 2.7.) This is because each part in the plan has a bad output and a bad input. In the explanation above of the tracing strategy, I pointed out that LOCAL could detect when it has closed a loop while tracing, but nothing was said as to how this state of affairs might be handled. Another specialist, LOOPS is invoked.

LOOPS<sup>1</sup> supplements LOCAL's hypothesis formation scheme based on tracing with other

---

<sup>1</sup> The simple feed-back configuration of figure 4.6 presents a situation which might seem to need special attention by LOOPS. Suppose the feed-back control signal were intended to have an amplitude that varied inversely with the forward signal's amplitude. Suppose further that the control signal is large in spite of the large forward signal. Though this is a bug within a loop, it is evidently findable by LOCAL without recourse to LOOPS.

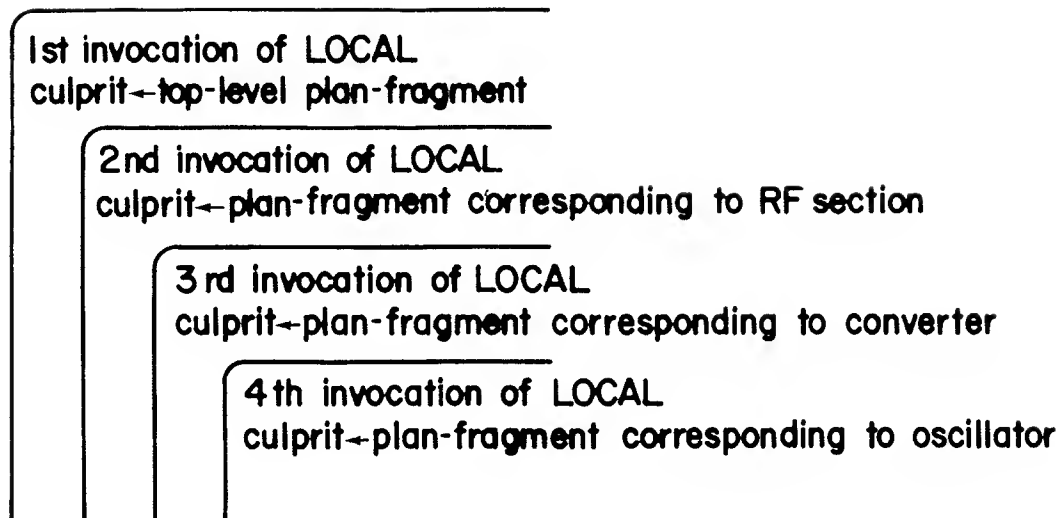


Figure 4.5 -- LOCAL's process state.

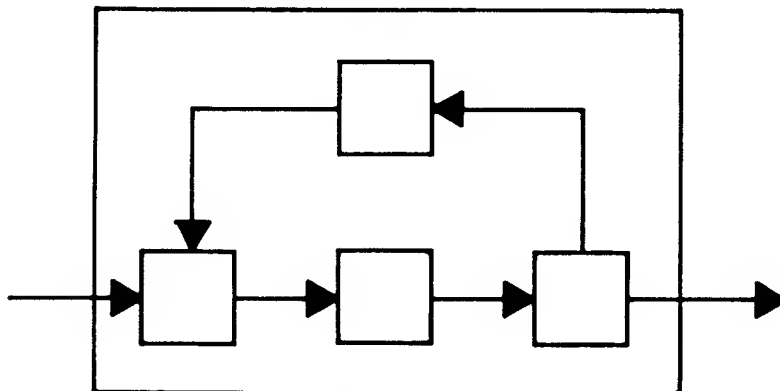


Figure 4.6 -- Three elements forward and one back.

schemes based on special knowledge about loop structures. LOOPS has two basic strategies. The first consists of looking at the power absorbing parts traversed in completing the loop and proposing failures based on knowledge of the purposes of those parts. The justification for this strategy is that powered parts have intrinsically higher *a priori* probabilities of failing because they encompass active components (and get hot), like transistors. The second strategy involves

breaking the loop by providing an externally (to the radio receiver) generated signal at some point in the loop. This signal must be appropriate to that point in the loop; i.e. a signal provided in breaking an AGC loop must look like the DC voltage that would be found there if the radio were working correctly. The loop breaking technique must isolate the failing part eventually (by tracing), but requires physical manipulations (which WATSON considers expensive), and the additional instrumentation necessitated by supplying the independent signal. The latter considerations cause LOOPS to use the first strategy preferentially.

Proceeding with the power-check strategy, LOOPS must first determine what the power absorbing plan-fragments are. Suppose PF-33 is a plan-fragment visited in the course of going around the troublesome loop. Evaluating

```
(FIND (THE-ONLY 1) (plan) '(PF PF-33 plan GR-78))
```

will yield the plan, AMP-34, of which PF-33 is an instance. AMP-34 is the plan of the amplifier used in the oscillator. Evaluating

```
(FIND ALL (port) '(PORT AMP-34 port ? POWER))
```

causes port to be bound to the name, in the plan AMP-34, of a power port. Finally, to determine how this port is actually instantiated in the design, evaluating

```
(FIND (THE-ONLY 1) (can-port) !"(PF-PORT PF-33 ,port can-port))
```

yields the canonical name of the realized port, PORT-35.

The next step is to propose a failure in AMP-34 that could plausibly explain the observed behavior of the oscillator. To do this LOOPS needs to know the *purpose* of the amplifier in the oscillator. This information can be obtained by evaluating

```
(FIND ALL (purpose) '(PF-PURPOSE (PF-32 PF-33 purpose))).
```

This returns a list of possible bindings of purpose, each element of which is a structure that looks like

(PF-PURPOSE PF-32 PF-33 predicate rule)

This says that the purpose of a particular part in a particular plan is to maintain the truth of some predicate via a computational rule. Specifically

```
(PF-PURPOSE PF-32 PF-33
  AMPLIFY-36
  (MAKE (> ~ (AMP-GAIN PF-32) 1.)))
```

says that PF-33 in PF-32 serves to keep a certain parameter of PF-32 greater than 1. The rule used to maintain this condition is AMPLIFY-36. WATSON says to himself, "Suppose the predicate of the purpose were not true." The only way this could happen is for the maintaining rule to be deactivated. What would the effect on the oscillator be if AMPLIFY-36 were turned off?

To answer this question, LOOPS must pull out yet another structure having to do with teleology. It is an item of the form

```
(PF-GOAL
  plan-fragment rule list-of-parts-involved
  predicate list-of-results)
```

Before examining the various slots in the PF-GOAL item and considering a specific instance of such an item, let's step back for a moment and consider what we want to accomplish with this structure. Each abstract plan is made up of parts, each of the latter serving a purpose in the overall plan. Recall that plans are used to realize parts in yet larger plans. The parts at the next level have purposes too. Understanding a design presupposes understanding how purposes at one level of planning get mapped into purposes at the next lower level. The oscillator is a part in the converter plan (refer to figure 2.6). Its purpose there is to maintain an oscillatory signal of fixed frequency and amplitude. How does an oscillator, composed of amplifier, tank and amplitude stabilizer serve this purpose? A PF-GOAL item explains all.

A PF-GOAL may refer to a particular list of plan-fragments in cascade inside a distinguished plan-fragment. The goal of this cascade is to maintain the truth of a certain predicate. This predicate will be the same as the predicate in some PURPOSE item associated with

a plan-fragment at the next level up in the design. The resulting structure is a kind of trace, or scenario, of the operation of the parts of the plan. The result list together with the rules mentioned in the RESULTS listed are a kind of control structure that represents the flow of causality in the plan mentioned in the PF-GOAL item. This trace reveals how the predicate of the PF-GOAL is established by the interaction among parts. Let's turn again to the oscillator that fails to oscillate. It is associated with the item<sup>1</sup>

```
(PF-GOAL PF-32 RULE-37
 (PF-33 PF-38 PF-39)
 (STABILIZE
  ,~(AMPLITUDE CARRIER SIGNAL OSC-OUT PF-32)
  (AT ~(IDEAL-AMPL PF-32)))
 (RES-40 RES-41 ... ))
```

The sub-plan-fragment list (the fourth position in this item) specifies the plan-fragments on the loop that lead to the present invocation of LOOPS. The result list (the sixth position) reveals what would happen if the amplitude of the output signal were less than the desired amplitude, ~ (IDEAL-AMPL PF-32), that is part of the design. The items

```
(RESULT RES-40
 (INIT)
 (< nowvalue ← ,~(AMPLITUDE CARRIER SIGNAL AMP-OUT PF-32)
  ~ (IDEAL-AMPL PF-32))
 (RESULT RES-41)
 (INIT)
 ampl1 ← ,~(AMPLITUDE CARRIER SIGNAL OSC-GAIN PF-32))
 (RESULT RES-42
 (INIT)
 ampl2 ← ,~(AMPLITUDE CARRIER SIGNAL OSC-OUT PF-32))
 (RESULT RES-43
 (AMP-LIMITER-LAW-44 RES-40)
 (> ,~(AMPLITUDE CARRIER SIGNAL OSC-GAIN PF-32) ampl1))
 (RESULT RES-45
 (AMPLIFY-36 RES-43)
 (> ,~(AMPLITUDE CARRIER SIGNAL OSC-OUT PF-32) ampl2))
 (RESULT RES-46
 (TANK-LAW-47 RES-45)
 (> ,~(AMPLITUDE CARRIER SIGNAL AMP-OUT PF-32) nowvalue))
```

-----  
<sup>1</sup> We shall eventually see that the structural relationship among PF-GOAL and PF-PURPOSE items is somewhat more complicated than the picture painted here. The items shown -- though not quite correct -- reveal the essential flavor of WATSON's analysis.



say that initially the tank input (refer to figure 2.7), tank output and limiter output take on values whose particulars are not important but they must be talked about, hence the Skolemizations `amp11` and `amp12`. The filter output has an amplitude less than the desired value. This results in an increased limiter output, hence an increased amplifier output (filter input), and finally an increased filter output. WATSON incorporates an analyzer for such scenarios (see chapter II) which is capable of detecting whether the prevailing rules (i.e. the ones active during the current activation of LOCAL) actually complete the scenario as specified by the result list of the PF-GOAL item. Other PF-GOAL items account for how stability is achieved when the output amplitude of the oscillator is high or on target. LOOPS reasons (via scenario completion) that if the rule, `AMPLIFY-36`, were deactivated, the results depending on it would cease to be true. An initial result in the structure indicates that the proper amplitude has been undershot. A final result (depending on `AMPLIFY-36`) states that the amplitude has been increased<sup>1</sup>, correcting for the initial state of affairs. Deactivating the rule, `AMPLIFY-36`, invalidates the result, leaving the amplitude in the initial state of undershooting the desired value, i.e. still at `nowvalue`. The result describing this latter state is matched against the actual state of affairs in the oscillator (Recall that it has zero output amplitude!) and is found to be consistent. LOOPS makes PF-33 the CULPRIT and calls LOCAL on it.

#### 4.4 The indictment of Q301.

The application of LOCAL to PF-33 introduces another twist. Until now the localization process has not explicitly mentioned any of the electrical mechanisms that underly the functioning of a radio. The plans that have been dealt with thus far have all been of type CASCADE. This makes it possible to think about radios in terms of sequences of abstract signal processors. Of course, MAXWELL knows that the magnitudes of certain voltages and currents (as obtained from

-----  
<sup>1</sup> An initial result is a given; that is, it is dependent on the INIT rule and no other result. A final result is one on which no other result depends.

various instruments) correspond to various abstract signal descriptors. Still, electricity has been pretty far removed from the fault-finding process. AMP-34 is a plan of type CIRCUIT. A CIRCUIT plan makes explicit the voltage/current interactions among its parts in the same way that a CASCADE plan makes explicit the signal interactions. Unfortunately, CIRCUIT plans do not have the quasi-causal<sup>1</sup> behavior of a CASCADE. Consequently it is unclear where to start doing either the forward or backward reasoning that is characteristic of tracing. There are, nonetheless, other powerful heuristics that may be applied. WATSON has already made use of the "unreliability" of plans that have powered parts. In this final phase of localization, he will again make use of that heuristic. He will also make use of the observation that the AC and DC behaviors of a CIRCUIT plan are largely decoupled and therefore may generally be analyzed separately.

LOCAL first asks if there are any power consuming plan-fragments in PF-33. Such a query produces Q301-48, a plan-fragment instantiating a plan of type COMPONENT<sup>2</sup>. Like other plans, COMPONENT plans describe a use (of an atomic electronic part). As such the COMPONENT plan for a bi-polar transistor used in a common emitter configuration is different from one used in a common collector configuration. Hence the two uses would be described by different COMPONENT plans.

As usual, the next step after producing a candidate is checking to see if the candidate's outputs are consistent with its inputs. This is the point at which DC and AC analyses separate. Almost any failure in an active component will lead to important changes in the prevailing biases. Another important observation is that when thinking about the DC properties of plan-fragments at the CIRCUIT plan level, "input" and "output" are not very meaningful, since any of the branch or node variables may be considered "independent." LOCAL will worry about DC consistency --

-----  
<sup>1</sup> A CASCADE plan embodies the sequential signal processing metaphor. Apart from the inconvenience of an occasional loop, this view of causality in a radio underlies the very powerful debugging tool, back-tracing.

<sup>2</sup> Note that the mapping from a design onto a circuit diagram is done by instantiating COMPONENT plans, giving the resulting plan-fragments names identifiable with components on the diagram.

consistency with node biases on the circuit diagram and/or the consistency of the measured DC conditions of a component with the components terminal voltage/current description. Evaluating

(TEST ' (DC-CONSISTENT Q301-48))

will reveal whether or not Q301-48 is in good shape with respect to its DC surroundings. Curiously enough, PF-33 is responsible for setting Q301-48's bias conditions (as is indicated by attached purpose commentary). The use of the JFET-49 plan, of which Q301-48 is an instance, requires that certain bias prerequisites be met. PF-33 takes responsibility<sup>1</sup> for meeting the obligation. It turns out that Q301-48's biasing is dependent upon the mode of operation of the oscillator. In particular, if the oscillator is not oscillating, the transistor will be biased for class A operation. Evaluating

(TEST ' (OSCILLATING PF-32)) ;PF-32 is the oscillator plan-fragment

during the course of the consistency check will inform LOCAL that the oscillator is not oscillating, hence class A biasing of Q301. Unfortunately, the biases on the circuit diagram correspond to measurements done with the oscillator oscillating. So comparing the actual state of the circuit with the diagram will not give a valid consistency check as it might if Q301 were being run as, say, a garden variety class A, common source amplifier.

All is not lost, however. In using the JFET-49 plan, the AMP-34 expects certain i/o behavior at the terminals of Q301. Q301 is an MPFI05 JFET. As such Q301 is quite accurately characterized by a certain table (which may be interpolated) of voltage/current measurements made at its terminals. LOCAL ADDs to the data base RESULTS that characterize the prevailing DC conditions:

```
(RESULT RES-50
  (INIT)
  (DC-BV (~ (GATE Q301-48) Q301-48 ~ (SOURCE Q301-48)) 0.)).
(RESULT RES-51
  (INIT)
  (DC-BV (~ (DRAIN Q301-48) Q301-48 ~ (SOURCE Q301-48)) 9.)).
-----
```

<sup>1</sup> In particular, PF-33 contains parts whose purpose is to achieve the correct bias conditions.

The ADDition of these facts triggers the firing of a number of rules. A KVL rule associated with Q301-48 computes the drain/gate branch voltage, and various rules associated with the nodes distinguished in PF-33 compute node voltages with respect to the declared ground, NODE-52. The products of these computations are RESULTS too. There is one more rule associated with Q301-48 that is of immediate importance, the one representing its voltage/current characteristics. On hearing the ADDition of the branch voltage consequents, this last rule, VIC-53, computes the branch current in the source/drain branch of Q301

```
(RESULT RES-54
  (VIC-53 RES-50 RES-51)
  (DC-BC (~ (DRAIN Q301-48) Q301-48 ~ (SOURCE Q301-48)) 9.)).
```

Finally, the KCL rule associated with the source node, ~ (SOURCE Q301-48), and the Ohm's law rule associated the plan-fragment for the source resistor, R302-55, compute respectively

```
(RESULT RES-56
  (KCL-57 RES-54)
  (DC-BC (~ (SOURCE Q301-48) R302-55 NODE-52) .007))
```

and (remembering that NODE-52 is the ground node)

```
(RESULT RES-58
  (OHM-59 RES-60) ; RES-60 is the result indicating
                  ; the branch voltage across R302.
  (DC-BC (~ (SOURCE Q301-48) R302-55 NODE-52) .02)),
```

both of which are claims about the quiescent branch current through R302. Consistency monitoring (see section 11.2) notes that RES-56 and RES-58 are incompatible. LOCAL is unleashed on Q301-48 which results in the transistor's being pulled for inspection. It is discovered that it has an opened gate.

## 5 The Case of C422

In this chapter I shall show much of the detail of how the scenario of the shorted capacitor is actually carried out by WATSON. There are two features of importance introduced here. The first is a technique that allows the consideration of non-linear signal processing in a very local fashion. The second is a new style of plan(-fragment) (whose type is called COUPLING) which engenders an important localization heuristic.

### 5.1 Getting started.

The complaint about the GR-78 is lodged by evaluating

```
(COMPLAINT GR-78
  (INPUTS
    (~ (OBS-SIGNAL PORT-1)
      (CARRIER-COMPONENTS
        ((MODULATION AM)
          (CARRIER-FREQ x)
          (AMPLITUDE z  $\in$  (L.10J *) | (r.10I *)))
        (MODULATION-COMPONENTS
          ((MODULATION-FREQ y  $\in$  (*)))))))
  (OUTPUTS
    (~ (OBS-SIGNAL PORT-2)
      (DISTORTION HARMONIC)
      (CARRIER-COMPONENTS
        ((MODULATION AM)
          (CARRIER-FREQ 0.)
          (MODULATION-COMPONENTS
            ((MODULATION-FREQ w  $\bullet$  z  $\leftarrow$  y |  $\in$  (~ y))))
            (MODULATION-AMPL v  $\bullet$  z  $\leftarrow$  0. |  $\in$  (> 0.)))))
  (CONTROL-BINDINGS
    (TUNING x  $\in$  (*)))
```

There are a number of new syntactic features introduced in this COMPLAINT. Notice the

declaration of the input signal amplitude,  $z$ . This says that  $z$  is to be some amplitude in the bottom ten or top ten percent of its allowable range, that is, a weak or a strong signal respectively. The features of interest in the output signal are specified by the declarations of the variables  $w$  and  $v$ . In particular

$$\dots w \bullet z \leftarrow y \mid \epsilon \{ \sim y \} \dots$$

says that the choice of a value for  $w$  depends on the way the value is chosen for  $z$ . If  $z$  is chosen from the generator  $\{L.10J \ast\}$  (i.e. weak signals),  $w$  is assigned the same value as  $y$ . On the other hand, if  $z$  is chosen from the generator  $\{r.10I \ast\}$  (i.e. strong signals), then  $w$  is chosen from the generator that produces harmonics of  $y$  (i.e. harmonic distortion).

$$\dots v \bullet z \leftarrow \theta. \mid \epsilon \{ > \theta. \} \dots$$

says that  $v$  is a zero or non-zero amplitude, depending on the choice of  $z$ .

The degrees of freedom that were noted in section 4.1 prevail here as well. An additional degree of freedom is introduced by the declaration of  $z$ . Recall that in section 4.1 I remarked that WATSON generally does not manipulate the amplitude setting of the signal generator. In this case he must do so since the COMPLAINT depends upon signal strength. In order to determine how to select this parameter for the validation process, WATSON needs to know something of the sensitivity properties of the GR-78. Looking into the data base, he finds

```
(PLAN RECEIVER CASCADE)
(PF PF-3 RECEIVER GR-78)
(PF-PARAM PF-3 SENSITIVITY REG-61)
(VALUE REG-61 .0000001)
(PF-PARAM PF-3 FRONT-END-OVERLOAD REG-62)
(VALUE REG-62 .00002)
```

These items give the upper and lower bounds for reasonable signals to be given the GR-78. WATSON takes the geometric means of .1 and 2.1  $\mu V$ , and 18 and 20  $\mu V$  respectively, yielding six possible set-points for the input signal amplitude.

These set-points are combined with the set-points developed in section 4.1 to generate

the configurations for the validation test, this time a loop with eighty-one inner rounds. The forward reasoning for validating the complaint proceeds essentially as before. The DETERMINER for the frequency of the modulation output component reveals that 1 kHz should be present. WATSON applies his frequency analyzer to the output port, PORT-2, of the GR-78 and duly notes the lack of a 1 kHz signal, corresponding to input signal amplitudes of  $2\text{ }\mu\text{V}$  or less, and the presence of 1 kHz fundamental plus significant harmonic components for input signals of amplitude  $10\text{ }\mu\text{V}$  or more. Having thus validated the complaint, a test set-up must be chosen for the localization process. WATSON is inclined to choose a test configuration from among those that reveal extraneous behaviors (the presence of harmonics in this case) rather than from those showing a lack of output<sup>1</sup>. So he chooses an input signal of 5 MHz carrier, 1 kHz modulation, and an amplitude of  $12\text{ }\mu\text{V}$ .

## 5.2 Localizing to the AGC.

He first queries the data base to see if he knows a bug that matches this situation. As before, the answer is negative and the tracing process begins. Successive applications of LOCAL reveal bad outputs from PF-19, and from PF-63, the plan-fragments corresponding to the RF section and IF strip respectively. SIG-MATCH fails at the output port of the RF section because MAXWELL reveals that there are several harmonics of the 1 kHz modulation, whereas the rule for PF-19 demands that only the fundamental of the 1 kHz modulation should be present. SIG-MATCH also fails at the output port of PF-63 because the carrier frequency of the observed output does not match the expected frequency -- 455 kHz. Analysis of the output signal reveals a 455 kHz carrier component plus a 1 kHz modulation with various harmonics of that modulation. Moreover, there are harmonics of the 455 kHz carrier. The audio frequency harmonics are not present in the input signal from the converter. Therefore the observed output does not match the

-----  
<sup>1</sup> There are likely to be many underlying causes for a lack of output, whereas, spurious output behavior may offer guidance as to the source of the wayward output.

expected output which has only the 455 kHz carrier and the 1 kHz modulation. The question then becomes whether or not the inputs to the IF strip are correct.

Proceeding to check the inputs of the IF strip, its main input, from the converter (see figure 2.5), is judged correct via forward reasoning facilitated by determiners. In attempting to match the auxiliary control input coming from AGC2 through port PORT-64 against the expected signal at that port, the control bias that is supposed to be developed there falls short of the expected value. This discovery triggers a *caveat alarm*<sup>1</sup> that informs LOCAL of the incorrect use of the plan, a token of which the IF strip (a part) is bound to. Usually the determination of the correctness of the output of a plan-fragment, given its inputs, is a straightforward use of determiners, rules, and signal back-tracing. In this case it is complicated by the fact that one of the inputs to the PF-63 directly contradicts a caveat, thereby failing to meet conditions imposed for the correct use of PF-63's type. Hence it cannot be guaranteed that the rules that usually describe the plan's input-output behavior remain valid. In order to find out what should happen in these adverse circumstances, WATSON must look inside PF-63.

The first order of business is to discover the source of the caveat, i.e. for what object in the design is the condition of the caveat being demanded? This is easily determined. When the caveat alarm went off, the variable *cave-canem* was bound to the name of the offended caveat.

### Evaluating

```
(FIND (THE 1) (pred dem) !" (CAVEAT ,cave-canem pred dem))
```

reveals the condition the caveat wanted to be true, i.e. the value of *pred*, and its source, i.e. the value of *dem*. *dem* is bound to PF-65, the plan-fragment which realizes the part of the IF strip called IF1 (see figure 2.10), an amplifier. In order to determine what the IF strip does when the condition of the caveat is not met, WATSON needs an explanation of the caveat. What is the nature of such an explanation?

-----

<sup>1</sup> The triggering is done via IF-ADDED methods. The declaration of a caveat in the assembly of a design (see chapter 10) engenders the creation of a collection of IF-ADDEDs that watch over the maintenance of the condition demanded by the caveat. When a contrary condition is asserted, LOCAL is interrupted and informed of the trouble.



In order to approach this last question properly, we need to understand better where the caveat came from in the first place and how it came to be visible immediately inside the plan for the RF section. PF-65, the plan-fragment corresponding to IF1, is an instance of an amplifier used in a class A regime. The type of this plan-fragment claims to have an externally supplied bias used to control the gain of the amplifier. The input-output design specifications associated with this type say that for control biases greater than 1.0 V it is a class A amplifier, for biases between 0.6 and 1.0 V it is a class B amplifier, and finally biases less than 0.6 V yield a class C amplifier. Consequently it is the use of this plan in the IF strip that is the source of the caveat. The caveat's becoming visible external to the IF strip plan-fragment, PF-63, inside the RF section is a result of the design assembly process. (See chapter 10 for further details.) PF-65 has a prerequisite called a *requirement*, which describes the bias to be supplied at PF-65's gain-control port that maintains its "class-A-ness." A requirement is similar to another class of prerequisites called *needs*. The former must be satisfied at "run-time" while the latter usually must be satisfied at design-assembly-time. In either case the satisfaction is guaranteed by the compilation and activation of IF-ADDED methods. In the present case, when MAXWELL ADDs to the data base the value of the bias measured at the port between the AGC2 and the IF strip, the method runs, reporting the problem to LOCAL. Note that though this method is compiled in the course of instantiating the plan of IF1 (yielding PF-65), this method in fact "listens" at the level of the IF strip. This is a consequence of the fact that the external bias port of the IF1 amplifier is also an external bias port of the IF strip. Hence, if there is trouble, it will be recognized at the first opportunity.

An explanation of the caveat, then, should allow WATSON to infer that IF1 is actually operating in a class C regime rather than the intended class A regime. WATSON first asks which of the possible uses of AMP2, the type of PF-65, matches the prevailing quiescent conditions surrounding PF-65 in the GR-78 circuit. The evaluation of

```
(OP-SPEC-MATCH 'PF-65 cave-canem)
```

yields a list of methods that give an imperative description of IFI. These methods embody the rule -- called AMPLIFY-66 -- and the determiners needed for predicting its input/output behavior under the prevailing circumstances. OP-SPEC-MATCH expects to find that the plan of which PF-65 is an instance has a number of alternative modes of operation. The caveat reflects having chosen one of those modes for use in the GR-78 design. The details of how OP-SPEC-MATCH does its job are tangential to WATSON's present line of thinking, so they will be postponed until section 10.7.

WATSON now has in his hands the behavioral description for IFI that actually applies -- in contrast to the description called for in the design. What does he do with it? He arrived at this point because he was attempting to rationalize the output of the IF strip in terms of its inputs. Ordinarily (apart from having set off the caveat alarm) he would simply have ground away with the necessary forward reasoning. Unfortunately, the rule for the IF strip relies on the rule for IFI. But WATSON has a new rule for IFI, included in the list of methods returned by OP-SPEC-MATCH. The IF strip, itself a CASCADE plan, is part of a CASCADE plan. Conceptually WATSON simply forgets (temporarily) the separate identity of the IF strip and inserts in its place the cascade inside the IF strip. The ability to do this transformation and use it to make sensible predictions relies on the fact that the signal descriptions generated by the rules for the IF strip's parts are compatible with the descriptions generated by the rule for the IF strip itself. Having thus *flattened* the IF strip modulo the IFI amplifier (a detailed account of which is given in section 11.4), WATSON carries out the following recipe:

1. Disable all the methods associated with PF-63 -- the plan-fragment for the IF strip.
2. Enable all the methods associated with the parts of the IF strip<sup>1</sup>.
3. Garbage collect all the items deduced using determiners that no longer apply.
4. Now do the forward reasoning to predict the behavior of the "phantom" IF strip.

-----  
<sup>1</sup> These would normally be activated when LOCAL was applied to PF-63. Turning them on inside the RF section is the essence of inserting the cascade composing the IF strip in its place.

Point '3' is technically complex, but not difficult, and depends on the use of the context machinery together with annotations on data base items indicating how they were formed in the first place.

Now WATSON can finish off the job of inferring that the mis-bias at the control port of the IF strip can cause the problem. At the output port of the IFI amplifier the forward reasoning just done predicts the presence of the first and second harmonics of the carrier, 455 kHz, each having a superimposed modulation consisting of the first five harmonics of 1 kHz<sup>1</sup>. The signal is propagated through the rest of the IF strip via the usual kinds of "linear" rules that we have seen before. SIG-MATCHing the predicted output of the IF strip against the output observed at the IF strip's output port completes a successful localization at this level.

### 5.3 Indicting C422.

At this point WATSON is pretty convinced that the source of the difficulty lies inside AGC2. Let's look into the circuit underlying AGC2 as illustrated in figure 5.1. It is a class B amplifier followed by an RC coupling network. The purpose of the circuit is to compute a time-averaged signal strength. The period with respect to which the time averaging is done is determined by the RC network of figure 2.12 and is about .1 seconds in this case. The mechanism by which this time averaging is achieved is not unlike demodulation. The class B amplifier rectifies the modulated signal, and the RC network does peak detection on the result, giving the largest value seen in the last .1 seconds. The class B amplifier supplies two other important functions as well. It is a buffer amplifier, thereby preventing excessive loading of the input to the detector of the GR-78. Also, being a common emitter configuration, it is inverting. Hence the

-----  
<sup>1</sup> The spectral analysis embodied in the AMPLIFY-66 rule is arbitrarily limited to thinking about the first two harmonics of the carrier and the first five harmonics of the modulation. This restriction is imposed because the presence of large harmonics components of this variety are sufficient evidence to indicate that observed outputs are explained solely by the inputs -- i.e. there is no problem inside IFI.).

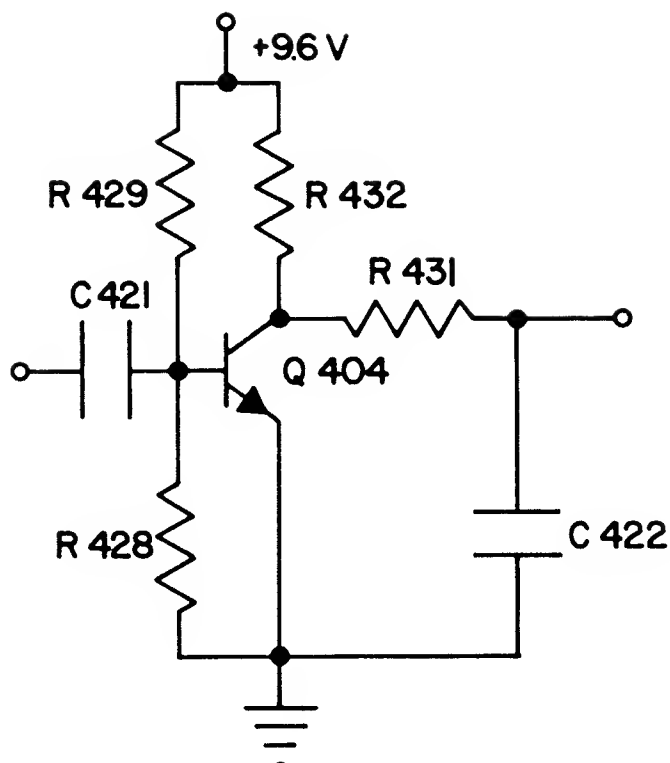


Figure 5.1 -- The circuit details of AGC2.

magnitude of the amplifier's output varies inversely with the size of its input. This is exactly what is wanted since the whole point of the AGC feed-back loop, of which AGC2 is a part, is to stabilize the strength of the signal delivered at the output of the IF strip in the face of any drifting in the signal strength at the input.

As I pointed out in section 1.4 the viability of tracing as a localization strategy relies upon the unilateral nature of the typical plan(-fragment). The RC network of figure 2.12 is not unilateral. In particular the voltage from A to ground affects and is affected by the voltage from B to ground. To embody this notion, among others, WATSON is aware of a class of plans I have not previously mentioned -- COUPLING plans. The essential feature of such plans is that they modify the relative concentrations of the various spectral components in their input signals.

Voltage measurements across either the input or output port<sup>1</sup> of such a plan would reveal identical signals modulo some DC offset. This latter feature suggests that it is meaningless to ask whether the output "follows" the input since they are not causally related. We will see shortly how this apparent bug in tracing can be used to WATSON's advantage.

To return to WATSON'S analysis, he sees the CASCADE plan of figure 2.11 and begins the trace as usual but notices that the first part on the signal path is bound to a COUPLING plan-fragment. Instead of verifying that the output from the RC network is that which would be determined by its inputs, he checks to see if the output is the same as the inputs modulo the DC offset associated with the plan-fragment. This can be done simply by SIG-MATCHING the two observed signals. The match fails because the output of the filter is a DC voltage whereas the input is a rectified, amplified and inverted copy of the input to AMP. So LO-PASS is not meeting its specifications.

LOCAL is applied to the plan-fragment, PF-67, to which LO-PASS is bound. From the point of view of hypothesis formation and forward reasoning PF-67 is a token of a CIRCUIT plan. Since this plan contains no powered parts (a general feature of COUPLING plans), DC analysis is not particularly interesting, hence only incremental analyses are carried out. Contemplating what part to propose as failing, LOCAL makes a choice based on *a priori* probabilities of failure. In particular capacitors are likely to fail before resistors. Further C422 is more likely to short than to open. WATSON hypothesizes the shorting of C422 and imagines what would happen if an incremental voltage increase were applied at A of figure 2.12 by ADDing

```
(RESULT RES-68
  (INIT)
  (AC-BV
    (~ (A PF-67) ~ (RES PF-67) ~ (CAP PF-67) ~ (GND PF-67))
    (UP (TOWARD 9.6))))).
```

The Ohm's law rule for R431 runs and ADDs

-----  
<sup>1</sup> The ports of such plans are generally voltage ports. The unwanted portion of the input signal is shunted into the local ground, letting the "good stuff" through.

```
(RESULT RES-69
(OHM-70 RES-68)
(AC-NV ~ (B PF-67) (UP (TOWARD 9.6))))).
```

If the capacitor C422 were doing its intended job, it would have frustrated R43I's attempt to pull up node B by hiding RES-69 and asserting

```
(RESULT RES-71
(DV/DT-72 RES-69)
(AC-NV ~ (B PF-67) UNCHANGED))
```

A similar analysis results if the voltage across the input port of LO-PASS were to go down incrementally.

WATSON's hypothesizing the shorting of C422 results in the disabling of the old rule, DV/DT-72, and the enabling of a new rule, CAP-BUG-RULE-73. Part of the enabling of this rule is the declaration of node B as an incremental ground. So if the same initial result as above were asserted in the context of the new rule, OHM-70 would assert that the current through R43I increases, but the node voltage at B is unchanged, which is precisely the symptom observed. C422 is pulled from the circuit and is discovered to be shorted.

## 6 A Case of Misalignment

In this chapter the details of the debugging of the misaligned front-end are revealed. In particular, the data structures underlying the concept of alignment are explained and WATSON's alignment expert, ALIGN, makes its debut. As with WATSON's other expert, LOOPS, ALIGN is brought to bear because of local plan structure and behavior.

### 6.1 The usual preliminaries.

WATSON is introduced to this new problem by telling him

```
(COMPLAINT GR-78
  (INPUTS
    (~ (DBS-SIGNAL PORT-1)
      (CARRIER-COMPONENTS
        ((MODULATION AM)
          (CARRIER-FREQ  $x \leftarrow *SOME$ )
          (MODULATION-COMPONENTS
            ((MODULATION-FREQ  $y \in \{*\}$ )))))))
  (OUTPUTS
    (~ (DBS-SIGNAL PORT-2)
      (CARRIER-COMPONENTS
        ((MODULATION AM)
          (CARRIER-FREQ 0.)
          (MODULATION-COMPONENTS
            ((MODULATION-FREQ  $y$ )
              (MODULATION-AMPL  $z \in \{> 0.\}$ ))))))
  (CONTROL-BINDINGS
    (TUNING  $w \leftarrow *SOME \neq x$ )))
```

The variable  $x$  is existentially specified as some particular carrier frequency. Any valid modulation frequency is assignable to  $y$ . The key feature to notice is that when the radio is tuned to some frequency  $w$  other than the broadcast frequency,  $x$ , the radio puts out a detectable audio

signal at the modulation frequency.  $y$ , as usual is assigned 1 kHz.  $z$ , being an output variable, will match any measurable amplitude. The assignments of  $x$  and  $w$  -- they being existentially specified -- are somewhat problematic. In principle, WATSON might have to examine all pairs of assignments to  $x$  and  $w$  in order to come up with an example that validates the COMPLAINT. In practice he takes the optimistic view that the existential specification indicates that validating values are easy to find. He arbitrarily sets  $x$  to 5 MHz and does a sweeping search downward for an appropriate value for  $w$ . The search is stepped at 100 kHz intervals, search stepping intervals and limits being chosen for convenience, keeping in mind the GR-78's receiving spectrum. At each of the receiver settings (the volume control being at its usual middling setting) WATSON carries out the usual SIG-MATCHes, and is successful at every tuning until 4.1 MHz, at which point he finds a significant 1 kHz component at the output. In particular with the generator broadcasting at 5.0 MHz, the receiver, tuned to 4.1 MHz, SIG-MATCH fails because there is observed a 1 kHz output when there should be none at all. Since he is in search mode, WATSON tries to optimize the find by making incremental variations in frequency around 4.1 MHz. This is done in increments of 10 kHz (because of channel width considerations imposed by civil law) and the amplitude of the 1 kHz output is maximized at 4.09 MHz. This configuration is the test set-up to be used in debugging.

## 6.2 Localizing the problem.

The back-trace proceeds as usual. SIG-MATCHing expected and observed signals reveals bad output from and input to the AF section (see figure 2.4). CULPRIT is therefore bound to PF-19, the plan-fragment for the RF section, and LOCAL is applied to it. Observation at the output of the detector (see figure 2.5) shows a 1 kHz modulation, which does not match the predicted output. But the input to the detector is a 455 kHz carrier modulated by 1 kHz, which by the rule for the detector transformation implies that the detector's output is reasonable. Similar reasoning applies to the IF strip. Its output, though wrong in principle, agrees with its input.



A caveat alarm is triggered because in the course of getting the observed input signal for the converter -- the plan-fragment for which is PF-74 -- from MAXWELL, a signal structure is generated having a carrier of 5.91 MHz. The alarm mechanism, as in the previous scenario, is the product of design teleology. As before, the sounding of this alarm binds cave-canem to the offended caveat and allows WATSON to find the demander of the caveat by evaluating

```
(FIND (THE 1) (pred dem) !" (CAVEAT ,cave-canem pred dem) )
```

which binds dem to PF-75, a token of the plan, MX, the plan for the mixer inside the converter. Evaluating

```
(OP-SPEC-MATCH 'PF-75 cave-canem)
```

returns a list containing the methods that actually govern the mixer's behavior in the face of images. As in the case of the IF strip, the converter is flattened (modulo the mixer) to account for its behavior in the presence of an image station. The recipe of section 4.3 again applies. Reasserting the observed sign descriptors at the phantom converter's input reveals that the image station should come through, exactly as observed. WATSON's attention turns to the apparent source of the 5.91 MHz signal, the RF amplifier.

The input to the RF amplifier is known to be correct since it comes from the signal generator. The structure underlying PF-76 is illustrated in figure 2.13. As usual, WATSON's inclination is to back-trace, but when MAXWELL is asked for the signal data, WATSON receives a polite refusal of the form

```
(CANT-MEASURE portname BECAUSE reason).
```

In particular the *reason* substructure (for the output filter) would be

```
(CHANGE ~ (BAND-PASS PF-77)) ;PF-77 is the plan fragment of the output filter
```

indicating that attempting to do measurements at the ports of the output filter changes its band-

pass characteristic<sup>1</sup>. It should also be pointed out, however, that even if the measurements expert had not complained, the plans associated with the input and output filters are of type COUPLING, hence not likely to be helpful in direct causal analysis.

Fortunately WATSON already noticed something that will prove helpful in the present circumstances. Recall that when he was looking into the cause of the caveat alarm, pred was bound to an explanation of why the alarm was sounded. The explanation is<sup>2</sup>

```
(FALSE
  (CARRIER-REJECTION
    (HIGH-BAND <<<.,~(FREQUENCY CARRIER SIGNAL OSC-IN PF-75)
      - .,~(FREQUENCY CARRIER SIGNAL RF-IN PF-75)> 2>
      + <.,~(FREQUENCY CARRIER SIGNAL RF-IN PF-75)>>
    (AT RF-IN)));
```

meaning that PF-75 expected no image signal to be at the port between the RF amplifier and the converter. The computation of the image carrier frequency embedded in this explanation of the caveat's cause for alarm yields a value of 5.91 MHz. WATSON asks the obvious question. Does PF-19 have anything to do with meeting the requirement implied by the caveat? Evaluating

```
(FIND ALL (rule part)
  '(PF-PURPOSE PF-19 part rule
    (CARRIER-REJECTION
      (HIGH-BAND <<<.,~(FREQUENCY CARRIER SIGNAL OSC-IN PF-19)
        - .,~(FREQUENCY CARRIER SIGNAL RF-IN PF-19)> 2>
        + <.,~(FREQUENCY CARRIER SIGNAL RF-IN PF-19)>>
      (AT RF-IN)));
```

yields an affirmative answer -- the RF amplifier plan-fragment (the value of part). Another FIND can be done to determine what parts in the RF amplifier support this goal. WATSON discovers that the input and output filters of the of the RF amplifier are active participants in the achievement of the goal.

-----  
<sup>1</sup> This message, incidentally, is a result of WATSON's reading commentary on PF-77 concerning its input impedance. As we shall later see, this is the same kind of commentary as is used by LOOPS when breaking loops.

<sup>2</sup> A pathname construct preceded by '.' is equivalent to a path name whose first path identifier is VALUE, i.e. .~(FREQUENCY ... ) is equivalent to ~(VALUE FREQUENCY ... ). Also, the bracketing characters, '<' and '>', are introduced to delimit subexpressions of the descriptive formulae used by WATSON. Since much of his descriptive notation is infix in nature, parenthesization distinct from that used for list structure is necessary.

### 6.3 Alignment.

Examining the plan of the RF amplifier reveals that a certain kind of prerequisite, called a *need*, must be met in order for the goal of image rejection to be met. In particular the need is one of alignment which will imagine for the time-being is specified by<sup>1</sup>

```
(RESULT PF-77
  (REQ)
  (ALIGNED
    ~(BAND-PASS PF-77)
    ~(BAND-PASS PF-78)
    STAGGERED)).
```

WATSON observes that this result is simply an *a priori* requirement for image rejection to work, so he asks if the requirement is met by evaluating

```
(TEST (ALIGNED ~(BAND-PASS PF-77) ~(BAND-PASS PF-78) STAGGERED)).
```

Testing the ALIGNED predicate has the effect of waking up the alignment expert, ALIGN, which will make it "as true as possible."

ALIGN must first compile an alignment plan for the GR-78<sup>2</sup>. He first finds all the parameters that could possibly need alignment by evaluating

```
(FIND ALL (aligant1 aligant2 alignotype)
  (RESULT ?
    ?
    (ALIGNED aligant1 aligant2 alignotype))).
```

Each of the returned bindings of aligant1(2) corresponds to a parameter of some plan-fragment.

The corresponding plan-fragments may be recovered by evaluating

```
(FIND (THE 1) (pf param-name)
  (PF-PARAM pf param-name ,aligant1))
```

for each alignment parameter. At this point WATSON is holding on to all the alignable parameters,

-----

<sup>1</sup> As usual, the real story is more complicated, as is revealed in section 10.5. In any event, meeting this requirement is necessary to fulfill the advertized purpose in the FIND of the last section.

<sup>2</sup> Alignment plans, like abstracted bugs, are associated with a design for later use.

their associated plan-fragments and the types of their alignments (staggered, tracking, etc.). He uses them to construct a cascade plan of the GR-78 flattened modulo these fragments. This construction is a straightforward recursive procedure. If at any level in the design hierarchy there is a plan-fragment whose type is CASCADE and any of whose parts is bound to a plan-fragment on the list of candidates gathered above, these latter cascaded plan-fragments are patched into the cascade structure at the position of the former plan-fragment. The flattening procedure ceases whenever ALIGN encounters a plan-fragment that is not of plan-type CASCADE, or whenever the list of alignment candidate plan-fragments is exhausted. The newly generated cascade is stored in the data base as the alignment plan for the GR-78.

ALIGN walks along the cascade created above, moving from RF section output to antenna input. Its first task is to take care of the oscillator in the converter. This step is motivated by the fact that the GR-78 is a superheterodyne receiver<sup>1</sup>. One of the alignment predicates discovered in the course of making the flattened cascade is

```
(ALIGNED
  ~ (BAND-PASS PF-32)
  ~ (BAND-PASS PF-78)
  TRACKING).
```

But it is also known that the band-pass of the RF amplifier's input filter -- whose corresponding plan-fragment is PF-78 -- is controlled by the receiver's tuning control. This fact is made evident to ALIGN by

```
(CONTROLLED-BY ~ (BAND-PASS PF-78) ~ (TUNING PF-3))
```

ALIGN therefore requests the signal generator to be set to broadcast at 7 MHz with a 1 kHz modulation. The receiver is tuned to 7 MHz. This broadcast frequency is chosen because it is a frequency sufficiently high to align the upper end of the tuning scale, while at the same time allowing room for an image to appear. In order to find out what control to adjust for alignment with respect to maximal frequencies, WATSON evaluates

<sup>1</sup> ALIGN knows that the most sharply tuned portions of RF circuits are usually modules involved in frequency conversion. Thus it focuses its attention on those modules first.

```
(FIND (THE 1) (tweakvar)
  (CONTROLLED-BY ~ (MAX-ADJUST BAND-PASS PF-78) tweakvar)).
```

The value of `tweakvar` is incrementally adjusted in the appropriate direction until the amplitude of the output signal of the mixer is maximized<sup>1</sup>. Similar maximizations are carried out for the band-pass parameters of PF-77 and PF-78, the output and input filters respectively of the RF amplifier.

At this point the first cut at aligning the GR-78 with respect to the high end of the receiver spectrum is complete. There is one possible bug, the oscillator may be offset in the wrong direction with respect to the input signal. ALIGN first calls for the signal generator to be set to maximum output amplitude, broadcasting at 7.91 MHz to make sure the image signal is properly rejected. The signal amplitude at the input to the detector should be way down with respect to the previously measured amplitude of the 7 MHz signal at this port. ALIGN then checks on the direction of the alignment by calling for the signal generator to broadcast at 6.09 MHz. If a strong signal should appear at the detector input, the sign adjustment of the oscillator above was wrong, ALIGN repeats the process, but adjusting the oscillator in the opposite direction. ALIGN then takes care of the low end of the spectrum by an analogous process making references to the data base looking for MIN-ADJUST commentary. ALIGN then repeats the minimal and maximal processes until no improvement is made at one end of the spectrum without making the other end worse. At the completion of this process, ALIGN looks at the symptomatic behavior that caused an interest in the RF amplifier, namely a large image component in its output. If the symptom has gone away, the motivating predicate test above will return FALSE and the bug is considered to be maximally localized. Otherwise TRUE is returned and LOCAL must look elsewhere for the source of the difficulty.

-----  
<sup>1</sup> The signal is measured at the output port of the mixer because that is the left-most measurable port affected by the alignment of the oscillator.

## 7 The Case of Q407

The complaint here is in fact the same as in the first scenario, hence lodged by telling  
WATSON

```
(COMPLAINT GR-78
  (INPUTS
    (~ (OBS-SIGNAL PORT-1)
      (CARRIER-COMPONENTS
        ((MODULATION AM)
          (CARRIER-FREQ)
            (MODULATION-COMPONENTS
              ((MODULATION-FREQ y € {*})))))))
  (OUTPUTS
    (~ (OBS-SIGNAL PORT-2)
      (CARRIER-COMPONENTS
        ((MODULATION AM)
          (CARRIER-FREQ 0.)
            (MODULATION-COMPONENTS
              ((MODULATION-FREQ 0.)))))
    (CONTROL-BINDINGS
      (TUNING x € {*}))).
```

WATSON performs the same validation tests as in the first scenario, resulting in a similar configuring of the receiver and signal generator for carrying out the failure localization.

### 7.1 Localizing to the audio driver amplifier.

WATSON's expectation is that the output of the GR-78 should have a 1 kHz modulation. Since it does not, LOCAL is applied to the top-level plan-fragment, PF-3, and begins tracing the cascade of figure 2.4. WATSON quickly discovers that the input to the AF section, whose corresponding plan-fragment is PF-20, is correct. In particular, the 1 kHz modulation is present. CULPRIT is bound to PF-20 and attention is focused on the AF section as expanded in figure 2.14.

SIG-MATCHING expected and measured signals at the interface of the audio controls plan-fragment, PF-79, and the audio amplifier plan-fragment, PF-80, reveals reasonable behavior as well. CULPRIT is rebound to PF-80, whose cascade structure is presented in figure 2.15.

Tracing along the cascade WATSON finds that the input to the power amplifier is incorrect, hence its quiescent output is justified. He moves on to the audio buffer amplifier, which is bound to the plan-fragment PF-81. As usual, MAXWELL is called on to reconnoiter the signal at the input port of PF-81. MAXWELL reads the commentary on PF-81 that says

```
(PLAN CC-AMP CIRCUIT)
(PF PF-81 CC-AMP GR-78)
(PF-PORT PF-81 IN PORT-82)
(PORT CC-AMP IN INPUT CURRENT),
```

indicating that the amplifier's information signal input port is a current port. MAXWELL knows that current port measurements are expensive since they entail breaking the circuit, properly terminating the two halves, and inserting an appropriate measuring instrument at the break. MAXWELL is obliged to warn LOCAL of an impending expensive measurement by GOing to the latter's measurement errors entry point with the message

```
(CANT-MEASURE PORT-82 WITHOUT (BREAK-AT PORT-82)).
```

WATSON wants to justify the output of the audio driver in terms of its input. He has just discovered that doing so in the obvious way will be costly. Perhaps justification can be achieved more cheaply? He notices that the audio driver is sandwiched between a plan-fragment whose input is known to be good (i.e. the audio pre-amplifier) and a plan-fragment (i.e. the audio power amplifier) whose output is known to be good given its inputs. At present the audio driver and audio pre-amplifier seem like equally good candidates for taking the blame for the observed misbehavior. WATSON would like a little evidence for blaming one or the other. So he asks whether there is any correlation between the audio pre-amplifier's output and its input. Using a current probe he SIG-MATCHes the output of the pre-amplifier with the signal generator running against that produced with the signal generator shut off. Visual inspection of an oscilloscope (by

the human assistant) reveals that the outputs are different in the two cases. No activity is observed at the output of the audio driver when the signal generator is running. PF-83, the plan-fragment for the audio driver is bound to CULPRIT and LOCAL is applied.

## 7.2 Proposal and partial rationalization.

PF-83 (see figure 2.16) is, of course, an instance of a CIRCUIT plan. To propose a candidate from among PF-83's parts, WATSON applies his standard heuristic of examining the power absorbing parts. This produces Q407-84, the COMPONENT plan-fragment that instantiates Q407. WATSON then looks to verify that its observed DC input/output properties meet specifications by examining the biases on the circuit diagram and comparing them with the observed values. The collector voltage of Q407 is observed to be down. Unlike the case of Q301 (see section 4.4), there is no good quantitative model for Q407's DC input/output behavior<sup>1</sup>, so WATSON needs more justification to blame the trouble on Q407.

As mentioned before, the rationalization process consists of proposing a failure mechanism, and separately checking if the predicted DC and AC consequences of the failure jibe with reality. WATSON knows that Q407, as a BJT, is most likely to fail by opening at one or the other of its junctions. Such a failure will change the rules of operation for Q407 as well as affecting the prevailing DC conditions. Looking at the open emitter/base junction failure first, WATSON finds out what such a failure entails by evaluating

```
(FIND (THE 1) (new-rule cons1)
  (FAILURE-MODE Q407-84 new-rule cons1)).
```

This yields a new governing rule (the binding of new-rule) to be observed by Q407 as well as a list of RESULTS, cons1, that follow from the failure. In a fresh context WATSON deactivates the old governing rule, activates the rule to which new-rule is bound, and ADDs the RESULTS on cons1.

<sup>1</sup> Q407 is a BJT. Consequently not much can be said about it except perhaps a minimum value of its  $\beta$ .



One of the RESULTS ADDED is

```
(RESULT RES-85
 (INIT-FAILURE)
 (DC-BC (~ (EMM Q407-84) Q407-84 ~ (COL Q407-84)) 0.)).
```

The KCL rule for the collector node runs and asserts that the current into R455 goes to zero

```
(RESULT RES-86
 (KCL-87 RES-85)
 (DC-BC (~ (COL Q407-84) PF-88 ~ (BOTTOM PF-88)) 0.)).
```

How did the KCL rule know that there was no current flowing into the base of Q408? At assembly (of the design) time the fact that Q408 is voltage driven was noted by its base node, hence the KCL rule knows that any current change out of Q407, will be reflected solely in current changes into the series resistance R455/R456.

Before going on with the other DC deductions to be made, some interpretation of the plan structures encompassed by the plan-fragments for the power amplifier, audio driver and audio pre-amplifier is in order. Figure 7.1 shows the parts involved in plan-fragment bound to the audio driver. PF-88 in RES-86 above is the plan-fragment corresponding to the collector resistance bias. The purpose of this part is to satisfy Q407's NEED for collector bias. The rule associated with this plan-fragment says that the voltage developed at the collector node is proportional to the current injected from the collector into the (series) resistance. The plan-fragment has two NEEDs of its own: that one of its terminals be an AC ground and that the other terminal have a source of DC current. The first is met by virtue of attachment to DC ground and the second is met by the transistor Q407. Consequently the assertion of RES-86 triggers the rule for the resistive bias plan-fragment which asserts

```
(RESULT RES-89
 (RBIAS-90 RES-86)
 (DC-BV (~ (COL Q407-84) PF-88 ~ (BOTTOM PF-88)) 0.)),
```

which indicates that the bias at Q407's collector has fallen to zero. Another part of DC interest in

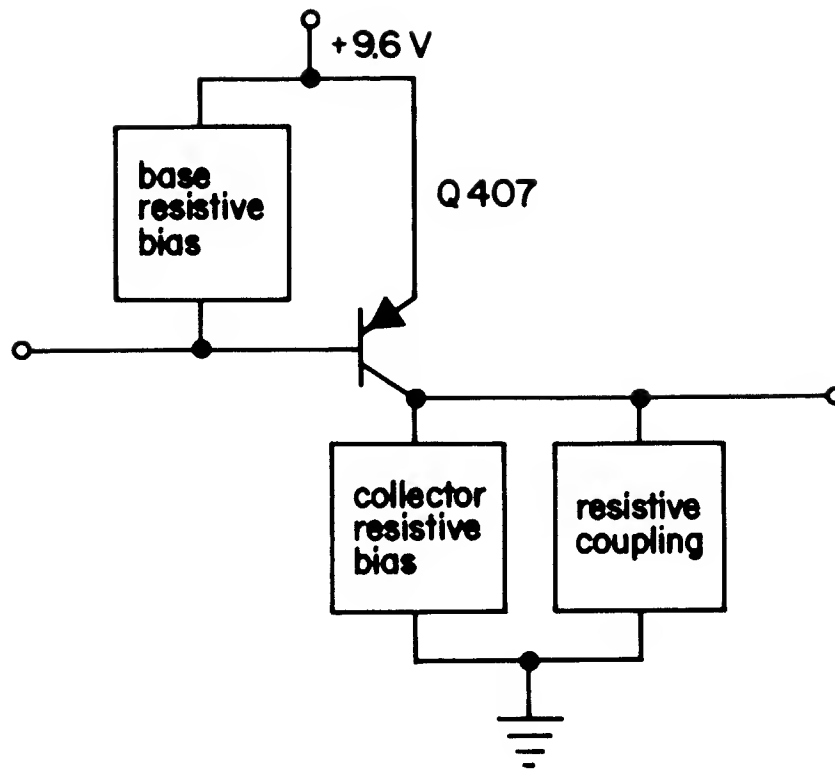


Figure 7.1 -- Expansion of the audio driver plan.

the plan of figure 7.1 is the base resistive bias which bound to a plan-fragment analogous to PF-88, the AC ground being the 9.6 V source and the current source being the transistor Q406 (not shown). There is also an associated rule that does computations analogous to those done by RBIAS-90.

Recall that the plan of figure 2.15 shows a DC feedback path. Moreover, the entire audio amplifier is a direct coupled circuit. As such the DC consequences whose computation we have just seen should lead to DC consequences in the audio power amplifier and the audio preamplifier as well. How is the computation of these consequences triggered? Figure 7.2 expands the audio power amplifier two levels of detail in terms of underlying plan-fragments. Figures 7.3 and 7.4 show the details of the two complementary amplifiers (shown in figure 7.2) that make the power amplifier work. In section 2.4 we saw WATSON deduce how various DC node voltages in

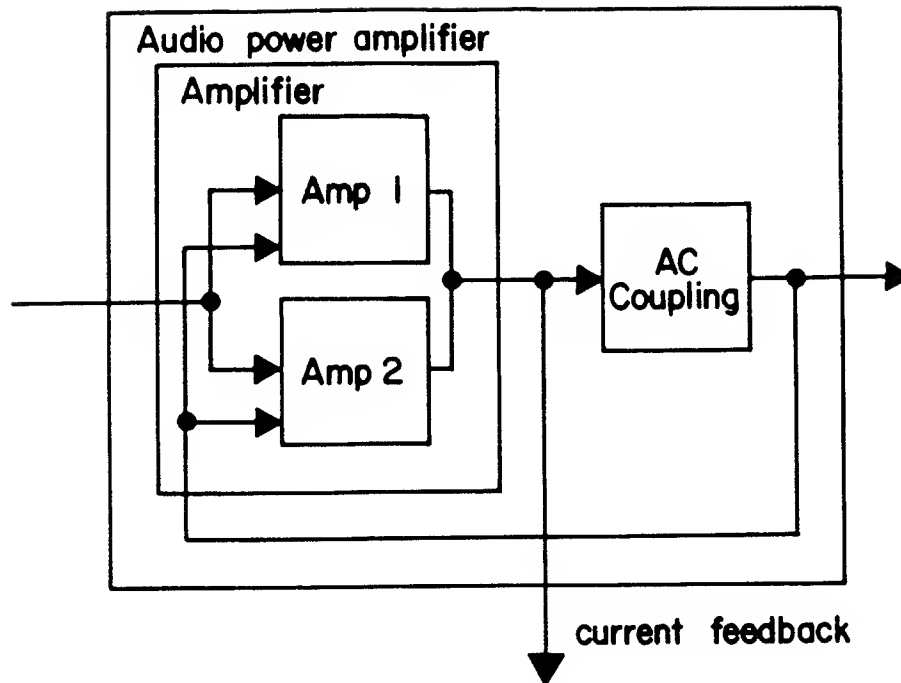


Figure 7.2 -- Double expansion of the audio power amplifier.

the circuit diagram of figure 2.16. We saw in the previous paragraph how some of these deductions were accomplished. We are now in a position to see how the others are done. Note that figure 7.3 shows a resistive bias part attached to the Q408 transistor. The plan-fragment corresponding to this part is PF-88, the very same plan-fragment as the one used to bias the collector of Q407<sup>1</sup>! This means that the bias changes computed by RBIAS-90 for the collector of Q407 are also changes for the base of Q408. Hence the base of Q408 falls to 0 V.

From figure 7.4 we can glean another bias deduction. Notice that there is a voltage divider bias element attached to the base of Q409. This part is bound to the plan-fragment VD-

-----  
<sup>1</sup> This illustrates an interesting feature of plan-fragments. They may be bound to parts in more than one super-plan-fragment. Notice that this is in addition to the possibility of a given plan's being instantiated in more than one plan-fragment.

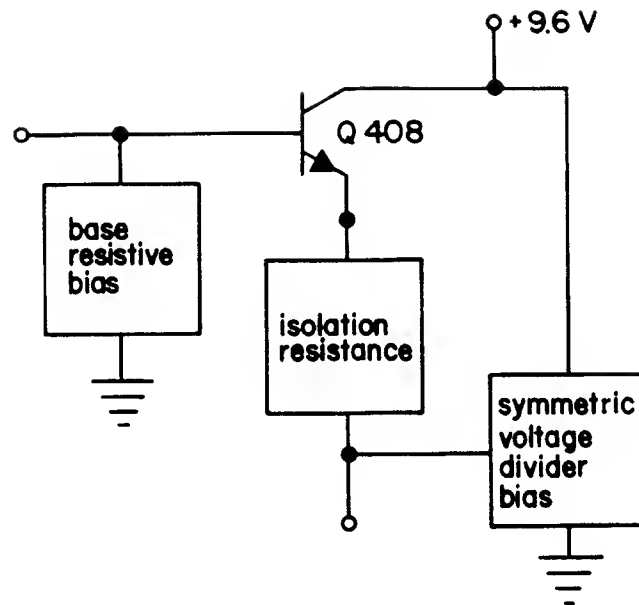


Figure 7.3 -- An expansion of Ampl.

91, an instance of the the voltage-divider plan. A voltage divider plan has two associated NEEDs, that its top and bottom terminals be incrementally at ground. The first NEED is met by PF-88 which biases the collector of Q407 (the top of the voltage divider)<sup>1</sup>. The second is met by the fact that the bottom of the voltage divider is determined at design assembly time to be at DC ground. The assertion of the RESULT RES-89 engenders

```
(RESULT RES-92
 (VDBIAS-93 RES-89)
 (DC-BV (~ (MID VD-91) VD-91 ~ (BOTTOM PF-88)) 0.))
```

asserting the collapse of the base of Q409 to DC ground.

-----  
<sup>1</sup> Not only can plan-fragments be used as parts within different super-plan-fragments, but they may also serve as external NEED satisfiers for various plan-fragments. Another feature of note, which is not made clear by the various figures, is that the same plan-fragments -- namely those bound to R455 and R456 -- underly both the PF-88 and VD-91 plan-fragments.

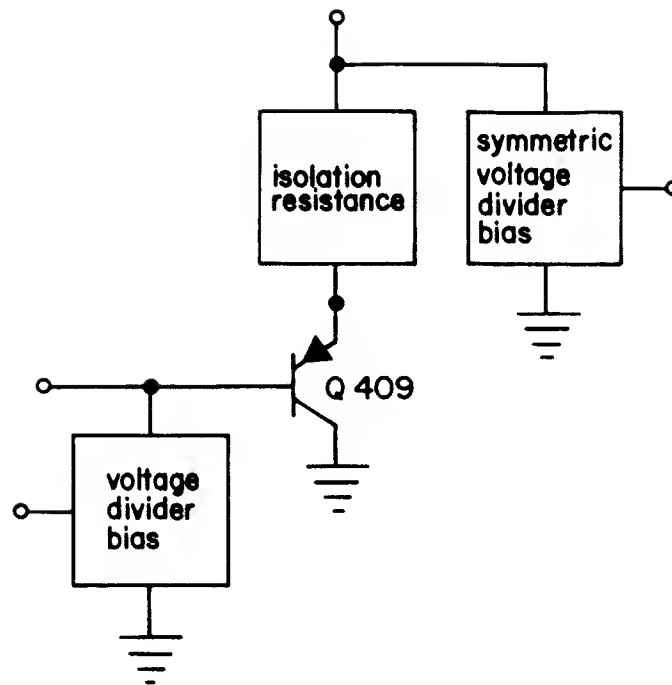


Figure 7.4 -- An expansion of Amp2.

### 7.3 More DC rationalization.

We have now seen the details of how the opening of Q407's base/emitter junction affects the node voltages on the output side of Q407. It remains to be seen how these effects are propagated to the output sides of Q408 and Q409. We want WATSON to deduce that NODE-94, the node called MID (between R457 and R458) in figure 2.16, falls to ground. To understand how this gets done we need to understand what the symmetric voltage divider (that is part of the plans of figures 7.3 and 7.4, does. This divider is a chain composed of Q408, R457, R458, and Q409. The divider is symmetric because Q408 and Q409 are a matched complementary pair. Q408 and Q409 are running in a class AB regime. That is, under quiescent conditions they are slightly

turned on. Because of C466 and R452, almost all the current coming out of the emitter of Q408 goes into the emitter of Q409. Symmetry demands then that the voltage drop between the collector of Q408 and MID be the same as the voltage drop between MID and the collector of Q409. The rule governing PF-95, the plan fragment bound to both instances of the symmetric voltage divider, says that as long as the current from the emitter of Q408 is the same as the current into the emitter of Q409, the voltage at MID is half the between the two collectors. A special case is noted when there is no emitter current at all. In this case the rule deduces that the voltage at MID goes to 0. Previous deductions have indicated that the bases of both Q408 and Q409 have been pulled to ground, leading the rules for Q408 and Q409 to conclude that those transistors' output branch currents are 0. This finally causes SYM-VD-96, the rule governing the symmetric voltage divider, to deduce

```
(RESULT RES-97
  (SYM-VD-96 ... )
  (DC-NV ~(MID PF-95) 0.)).
```

Thereafter FDBK-98, the rule for the current feedback network of figure 2.15 deduces

```
(RESULT RES-99
  (FDBK-98 RES-97)
  (DC-NV ~(EMM Q406-100) (DOWN (TOWARD 0.)))) .
```

Various rules then lead to the deduction of the increased branch current in the output branch of Q406 and an increased branch current across R454, just as claimed in section 2.4. At this point all of the DC deduction propagation has died out, and WATSON must prepare predictions with reality. Unfortunately the deduction

```
(RESULT RES-101
  (OHM-102 ... )
  (DC-BV
    (~ (TOP R454-103) R454-103 ~(BOT R454-103))
    (INCREASED (FROM 0.3))))
```

indicating an increased voltage drop across R454 disagrees with observation. As remarked in

section 2.4, this contradiction invalidates the hypothesis of the open emitter/base junction. Analogous reasoning to that we have been following, but predicated upon the opening of the collector/base junction, does, however, lead to DC predictions consistent with observation.

#### 7.4 AC rationalization.

At this point the hypothesis of the open collector/base junction looks very good to WATSON as the underlying cause of the lack of signal at the output port of the audio driver amplifier. WATSON became suspicious of the audio driver in the first place because an apparent signal at the input port lead to no interesting activity at the output port. Translated into the terms of voltages and currents appropriate to CIRCUIT plans, the question to be asked is "What effect does an incremental current change at the input port of the audio driver have on its output (voltage) port?" Actually, one should ask about both incremental increases and decreases, and to that end WATSON first asserts

```
(RESULT RES-104
  (INIT)
  (AC-BC (~ (BAS Q406-105) Q406-105 NODE-52) (DECREASED (TOWARD 0.)))).
```

The firing of various rules for the pars of PF-83 eventually conclude with

```
(RESULT RES-106
  ( ... )
  (AC-BV (~ (COL Q407-84) PF-88 NODE-52) UNCHANGED))
```

which says there is no effect on the output port of the audio driver. Similar deductions leading from the assertion of increased input current also lead to a conclusion of unchanged output. This completes the AC rationalization and also convinces WATSON that Q407 (being bound to a COMPONENT plan-fragment) is the underlying cause of the problem. Pulling it from the circuit and examining it indicates that it is indeed broken.

## 8 Bug Abstraction.

Recall that bug abstraction is the process whereby WATSON characterizes the untoward behavior of a given plan-fragment in terms of symptomatic behavior observed in one of its parts (the latter being realized as plan-fragments too). The principal technical issues here are two: when is abstraction appropriate, and how is the most general description of the bug to be obtained? The first issue follows from the observation that localizations achieved by LS1 and LS2 are always easy to find again. Localizations achieved by other strategies are hard because the forward reasoning that rationalizes a hypothesized part failure is usually complicated, hence it would be convenient to avoid such rationalization on future encounters with the same bug.

The second issue arises from the fact that the bug, whatever it is, is observed in a particular design. The alignment bug (chapter 6), for example, is a possible bug of the GR-78. The abstract description of the bug is: "The appearance of the same station at two places on the dial might be due to the misalignment of the RF section." We note, however, that this bug is not only possible in the GR-78, but in any receiver whose RF section is designed according to the plan of figure 2.5. So it seems reasonable to achieve generality by abstracting bugs not with respect to designs (or plan-fragments), but rather with respect to plans. In this chapter we will see how the "interesting" bugs are abstracted with generality.

### 8.1 Q301.

The criterion prompting an abstraction is that a proposed failure require rationalization by forward reasoning. In the case of Q301 (chapter 4) all localizations are products of LS2 until



arriving at the oscillator plan-fragment, PF-32. Consequently those localizations will not engender abstraction. Recall that WATSON had to demonstrate to his satisfaction that the failure of the amplifier could result in the observed lack of oscillation. So localizing from the plan-fragment to the amplifier plan-fragment required forward reasoning, hence prompts abstraction. Localizing from the amplifier plan-fragment to its contained transistor set off forward reasoning that confirmed the incorrectness of the observed bias. This localization will not be abstracted, despite the forward reasoning. Bugs at the circuit plan level are not abstracted. This weakness in WATSON is a result of my presently deficient understanding of the "right" representation of circuit plans<sup>1</sup>.

The informal description of the product of abstraction for the oscillator plan is: If the oscillator is not oscillating and the amplifier part of the oscillator has an active component whose bias is incorrect, then more than likely the oscillator's trouble is to be found in the amplifier<sup>2</sup>. A formal description of the abstraction is<sup>3</sup>

```
(BUG B-107 OSC
  (SYMPTOMS pf (OUTPUTS (~(OSC-OUT pf) sig)))
  (SIGNS part . sign)
  (MISSING-PURPOSES . mp)
  "AUX" (sig-108 (part (PATH-EVAL !" (AMP ,pf)))
  (GEN-SIGNALS
    (OSC pf
      (OUTPUTS
        (sig-108 = OSC-OUT
          (CARRIER-COMPONENTS
            ((MODULATION NIL)
              (AMPLITUDE 0.)
```

<sup>1</sup> The present representation of circuit plans does not facilitate the abstraction of bug descriptions of sufficiently general applicability. I shall have considerably more to say about this problem in chapter 12.

<sup>2</sup> Notice that this abstraction includes in the sign details of a part's realization in the sign. In general, abstraction will not require looking inside the parts of the plan over which the abstraction is being done. Recall, however, that LOOPS was invoked because each of the oscillator's parts seemed to be delivering reasonable outputs in light of its inputs. Consequently no sign for the abstraction can be obtained from descriptions of the extrinsic performance of the parts of the oscillator plan. Hence the sign must be developed from the next level of localization.

<sup>3</sup> Note that the function PATH-EVAL represents an explicit invocation of the function called implicitly by the path evaluation macro characters.

```

(CARRIER-FREQ x € (*))))))
(CONTROL-BINDINGS
 (TUNING-FREQUENCY x)))
(COND ((SIG-MATCH sig-108 sig)
 (COND ((AND
 (LOOPING?)
 (FOR (AT-LEAST 1) (part1) !" (ACTIVE-PART-IN ,part ,part1)
 (COND ((NOT (DC-CONSISTENT part1))
 (NOTE
 !" (BUG B-107 OSC
 (SYMPTOMS ,pf
 (OUTPUTS (~ (OSC-OUT ,pf) ,sig)))
 (SIGNS ,part . sign = (
 (TEST
 (AND
 (LOOPING?)
 (FOR (AT-LEAST 1) (part1)
 !" (ACTIVE-PART-IN ,part1 ,part)
 (COND ((NOT (DC-CONSISTENT part1))
 (RETURN T))))))))))
 (MISSING-PURPOSES . mp = (
 ,part
 (AMPLIFY
 (BETWEEN
 , , ~ (OSC-IN ,part)
 , , ~ (AMP-OUT ,part))
 (SET-BY
 , , ~ (AMP-GAIN ,pf))))))))))))))

```

The first thing to be noticed is that a bug abstraction is a piece of code whose invocation is directed by the pattern

```
(BUG ... (SYMPTOMS ... ) (SIGNS ... ) (MISSING-PURPOSES ... )).
```

This seems like a natural solution to the abstraction problem, because the determination of symptom/sign match is not just a matter of matching fixed structures. The structures to be matched must be constructed at "run-time" in order to achieve the requisite generality. Hence some code must be generated for execution at run-time to make this dynamic determination.

When the code body for the B-107 (the BUG's name) is reached, the pattern matching variables *pf* and *sig* have been bound to the arguments supplied by the invoker (see section 4.2). An auxiliary variable, *sig-108* is declared, and will eventually become the repository for a signal structure created by a call to GEN-SIGNALS. (GEN-SIGNALS ... ) describes and produces a

signal structure to be associated with the output port (by ... sig-108  $\equiv$  OSC-OUT ... ) of the oscillator the plan. The syntax of GEN-SIGNALS is much like that of a COMPLAINT. In the present case it describes a signal structure that is an unmodulated carrier of negligible amplitude, with a carrier frequency that may take on any legal frequency for the oscillator instantiated as the value of pf. In the scenario explained in chapter 4, an attempted match for an already abstracted bug would complain that the oscillator shows negligible output amplitude at the carrier frequency of 5.455 MHz. The bug abstraction exhibited above generalizes (among other things) the frequency of the oscillator (and would match the COMPLAINT). How is the more general statement obtained? Specifically, how is the carrier frequency descriptor variabilized?

The central idea is that a run-time environment is maintained that "remembers" how signal descriptors arose<sup>1</sup>. The bug abstracting machinery has access to the results of the signal match that reported the mismatch at the output port of the oscillator. The essential details of that mismatch are seen in the following pairing of observed and expected signal structural components:

```
(PF-OBS-SIGNAL PF-32 PORT-109 SIG-110)
(PF-SIGNAL      PF-32 PORT-109 SIG-111)
(CARRIER-COMPONENT SIG-110 CC-112)
(CARRIER-COMPONENT SIG-111 CC-113)
(MODULATION CC-112 MOD-114)
(MODULATION CC-113 MOD-115)      (DETERMINED-BY MOD-115 GIVEN)
(VALUE MOD-114 NIL)
(VALUE MOD-115 NIL)
(CARRIER-FREQ CC-112 CF-116)
(CARRIER-FREQ CC-113 CF-117)    (SET-BY CF-117 ~(TUNING-FREQUENCY PF-32))
(VALUE CF-116 5455000.)
(VALUE CF-117 5455000.)
(AMPLITUDE CC-112 AMPL-118)
(AMPLITUDE CC-113 AMPL-119)
(VALUE AMPL-118 0.)              ;Mismatch begins here.
(VALUE AMPL-119 1.4)             (ELEMENT-OF AMPL-119 {> 0.})
```

The mismatch, of course, occurs in the last two lines. The amplitude of the observed signal is 0, whereas the predicted signal must be chosen from a generator that excludes 0, as is indicated by

-----

<sup>1</sup> The inspiration for this technique is again Winston's [Winston, 1975] c-notes.

the ELEMENT-OF commentary for the ob AMPL-119. To generalize the description of the symptom, WATSON looks first at the values of obs in the predicted signal that do match the analogous values for the observed signal. The commentary on the modulation indicates that it is a "given," hence not variabilizable. The commentary on the carrier frequency, (SET-BY ... ), says that it is determined by some control parameter of PF-32. This commentary is a kind of "indirect" pointer to the precursors of the carrier frequency. This prompts WATSON to begin a process of "pointer-chasing" to find the ultimate source of the carrier frequency's value. On  $\sim$ (TUNING-FREQUENCY PF-32) -- which denotes the ob, REG-120 -- is found the annotation

(CTL-OFFSET-WRT REG-120  $\sim$ (TUNING PF-74) (BY  $\sim$ (IF-FREQ PF-74)))

indicating that the frequency of the variable-frequency oscillator is identified with an external control of its super plan-fragment, PF-74, which you will recall is the plan-fragment for the converter. However, the value of the frequency parameter does not correspond identically, but is offset by another parameter, the IF frequency associated with the converter plan-fragment.

A chain of indirect pointers connecting plan-fragment parameters occurs and may be "chased" to its terminus at the station-tuning parameter of the top-level plan-fragment of the GR-78. Let this last parameter be represented by the ob, REG-121. WATSON finds the commentary

(CHOSEN-FROM REG-121 (\*)),

indicating that the value of the parameter could potentially be any legal tuning setting. This commentary was generated at the time of evaluating the COMPLAINT of chapter 4, which indicated the GR-78's failure to produce audible output no matter what station it was tuned to. This tells WATSON that he can variabilize the tuning parameter (with respect to the mentioned generator, of course). He carries the variabilization back down the chain of indirection until the CTL-OFFSET-WRT identification is reached. Let's think of the variable associated with the parameters on the chain as  $y$ . When WATSON reaches the ob with the CTL-OFFSET-WRT commentary, he manufactures a new variable,  $x$ , which he declares equal to

$\langle y \in \{*\} \rangle + \sim$ (IF-FREQ PF-32),

which can be rearranged to

$$\langle y + \sim(\text{IF-FREQ PF-32}) \rangle \in \{*\}.$$

WATSON simplifies the last to the  $x \in \{*\}$  declaration seen in the GEN-SIGNALS subexpression of B-107s code body<sup>1</sup>. The rest of descriptor values assigned in the GEN-SIGNALS are produced simply by copying the values from the expected signal structure.

The "business end" of the code body -- (COND ...) -- checks first to see if the reported signals of the instantiation of the oscillator plan are like those abstracted in the BUG. If this match is successful, the sign of the BUG is compared with that seen in the prevailing circumstances, eventually producing a candidate broken part (the binding of part) if the situation warrants it. How was the (COND ...) produced? I have already mentioned that LOCAL has a local variable, culprit. It is bound by virtue of a parameter passed in the calling of LOCAL. What I have not mentioned is the fact that other parameters are passed by the call. The parameter of interest here is reason, whose value is a structure that justifies having put the blame on the plan-fragment to which pf is bound. We shall see how reason is used presently.

The value of reason is a list of tests to be performed. Their success justifies assigning blame. Recall that WATSON invoked LOOPS because he traced his way around a looping signal path, having noted that each part on the path has reasonable outputs given its inputs. When activated, LOOPS first sought a power consuming part whose failure might explain the observed lack of oscillation. Unfortunately there is no identifiable sign to be found in the oscillator plan that would indict the amplifier as the culprit. Such an indictment can be made only by carrying out one more level of localization -- another invocation of LOCAL -- to find a problem inside the amplifier. To obtain a plan-fragment to apply LOCAL to, a search was done inside the amplifier for active elements that do not meet their DC specification. Keep in mind that we are

-----  
<sup>1</sup> WATSON determines the ob  $\sim(\text{IF-FREQ PF-32})$  to have an unvariabilizable value, i.e. a constant, because of the pointer-chasing similar to that just done. The chain in that case will terminate at an ob whose value is fixed by the design assembly process.

considering several levels of localization: (LOCAL PF-32 ... ), (LOCAL PF-33 ... ), and (LOCAL Q301-48 ... ). The reason for applying LOCAL to Q301-48 was that it was discovered to be failing to meet its DC specifications. Ordinarily the abstraction of the bug occurring between the first and second levels of LOCAL's invocation cannot see the binding of reason for the third invocation. But because this last occurs in the scope of an invocation of LOOPS, its reason is preserved for use at the higher level. The method of preservation is to APPEND it to the previous binding of reason. The code body composed will first execute the AND of all the tests on the reason list. The success of these tests will cause the second part of the code body to run, which causes the invoker of the abstracted BUG to be presented with an instance of the abstraction of the form

```
!" (BUG B-107 OSC ,pf ,part
    (SYMPTOMS ... )
    (SIGNS ,pf ,sign)
    (MISSING-PURPOSES . ,mp))
```

which corresponds to the invoking pattern. If the tests do not work out no BUG is NOTEd.

The (MISSING-PURPOSES ... ) clause of the NOTEd bug documents something of the mechanism of the failure, namely what purpose(s) are not being fulfilled, leading to the observed failure. Yet another parameter passed on calling LOCAL is clobbered-purpose. Recall that in the scenario Q301, a goal for the oscillator was recovered indicating how it was to stabilize its amplitude. As we shall see in chapters 10 and 11, plan goals are affiliated with various parts and purposes for the parts. Hypothesizing a broken amplifier suggests that the amplifier is not fulfilling its amplification purpose in the scheme to stabilize the oscillator. The unfulfilled purpose is recorded in clobbered-purpose. The name of the plan-fragment whose purpose is unfulfilled is the plan fragment to which the failure is next localized.

## 8.2 C422.

In the case of the AGC bug caused by the failure of C422, the problem is successively

localized by tracing (LS2) until arriving at the RF section. Here too the problem can be further localized by LS2 since the output of AGC2 is not reasonable given its inputs. However, the violation of a caveat prompts forward reasoning, and in turn, bug abstraction<sup>1</sup>. In any event, this is the only abstraction done, and it results in

```
(BUG B-122 RF-SECT
  (SYMPTOMS ,pf
    (INPUTS (~(ANT-IN pf) sig1))
    (OUTPUTS (~(OET-OUT pf) sig2)))
  (SIGNS part . sign)
  (MISSING-PURPOSES . mp)
  "AUX" (sig-123 sig-124 sig-125
        sig-126 (part ~ (AGC2 pf)))
  (GEN-SIGNALS
    (RF-SECT pf
      (INPUTS
        (sig-123 = ANT-IN
          (CARRIER-COMPONENTS
            ((MODULATION AM
              (CARRIER-FREQ r ∈ {*})
              (AMPLITUDE s ∈ (f.101 *))
              (MODULATION-COMPONENTS
                ((MODULATION-FREQ t ∈ {*})))))))
        (OUTPUTS
          (sig-124 = OET-OUT
            (CARRIER-COMPONENTS
              ((DISTORTION HARMONIC)
               (MODULATION NIL)
               (CARRIER-FREQ 0.)
               (MODULATION-COMPONENTS
                 ((MODULATION-FREQ u ∈ {~ t})))))))
        (CONTROL-BINDINGS
          (TUNING r)))
      (AGC2 part
        (INPUTS
          (sig-125 = IF-DET
            (CARRIER-COMPONENTS
              ((DISTORTION HARMONIC)
               (MODULATION AM)
               (CARRIER-FREQ u ∈ (~ ~ (IF-FREQ pf)))
               (AMPLITUDE x ∈ (> 0.))
```

<sup>1</sup> An argument might be made for doing another level of abstraction inside the plan-fragment for AGC2 which would map the AGC's failure into a failure in the low pass output filter, LO-PASS. Recall that localization to the filter was done by a variant on tracing which checks to see whether the input and output of the filter are the same modulo a DC offset. Though WATSON must, in fact, look inside the filter (to the extent of finding out whether or not it is realized by an instance of a COUPLING plan) it seems hardly worth distinguishing this heuristic from run-of-the-mill tracing, hence not worthy of abstraction.

```

      (MODULATION-COMPONENTS
        ((MODULATION-FREQ w))))))
(OUTPUTS
  (sig-126 = AGC2-BIAS
    (CARRIER-COMPONENTS
      ((MODULATION NIL)
       (CARRIER-FREQ 0.)
       (AMPLITUDE y € (≠ ~ (CLASS-C-BIAS IF-STRIP pf)))))))
(COND ((AND (SIG-MATCH sig-123 sig1) (SIG-MATCH sig-124 sig2))
  (COND
    ((AND
      (SIG-MATCH sig-125 ~ (SIGNAL IF-DET pf))
      (SIG-MATCH sig-126 ~ (SIGNAL AGC2-BIAS pf)))
      (NOTE
        !" (BUG B-122 RF-SECT
          (SYMPTOMS ,pf
            (INPUTS (~ (ANT-IN ,pf) ,sig1))
            (OUTPUTS (~ (DET-OUT ,pf) ,sig2)))
          (SIGNS ,part
            (INPUTS (~ (IF-DET ,pf) ,sig3))
            (OUTPUTS (~ (AGC2-BIAS ,pf) ,sig4)))
          (MISSING-PURPOSES . mp = (
            ,part
            (PROVIDES
              (BIAS (ACROSS , , ~ (AGC2-BIAS ,pf)))))))))).

```

This abstraction includes a symptom -- a description of signals at ports of an instantiation (pf) of the RF section plan, and a sign -- a description of signals at ports of a part of the RF section plan. The various ports in the plan and its parts have variables associated with their respective signal structures (by the '=' operator) so as to facilitate referencing these structures in the BUG's subsequent program body. There are several variabilizations in the bug abstraction: r, s, t, u, v, w, x, and y. We shall now look into the methods used to create them.

First, considering the signal at the antenna port of the RF section, we observe that the assignment of the modulation descriptor is a constant, AM. It is a constant because the expected and observed signals seen at this port by LOCAL were found to match (at this parameter), and the value for this parameter in the expected signal originated in the design. The carrier frequency is variabilized as 'r' and is declared to be able to take on any legal frequency value. This variabilization is a consequence of the fact that the antenna port of the RF section is an external port. That is, it is the same as an input port of its super plan-fragment. WATSON recalls that the



signal at this port was determined by the COMPLAINT that led to the localization of C422. In that recollection he notes (via CHOSEN-FROM commentary) that *r* was free to be chosen in this way. The variabilizations 's' and 't' are similarly recollections of the COMPLAINT validation process. The first variabilization is accounted for by having chosen the modulation from the upper ten percent of the available frequencies -- '{f.101 \*}'<sup>1</sup>. The second variabilization recalls the arbitrary choice of modulation frequencies.

The only variabilization in the signal at pf's audio port is the modulation frequency. We shall examine the construction of the variabilization momentarily. Apart from the DISTORTION descriptor, all the other signal descriptors are constants arising from the success of the SIG-MATCH between expected and observed signals modulo those descriptors. The distortion descriptor is generated because SIG-MATCH failed at matching expected and observed modulation components. The mismatch is due to the presence of harmonics of the 1 kHz fundamental. The generalization of the signal modulation descriptors' selectability from harmonics of 1 kHz to harmonics of an arbitrary audio frequency proceeds by ancestral analysis of the expected frequency parameter. This frequency is determined by forward reasoning through the firing of the rules associated with the various parts of the RF section. When such rules fire, they will frequently comment value results that they produce. Suppose the ob corresponding to the frequency descriptor of the modulation (in the expected signal) is REG-127. When the rule caused the appearance of the proposition

(VALUE REG-127 1000.)

(indicating a 1 kHz modulant), it also commented the result by recording

(RULE-IDENTIFIED-WITH REG-127 REG-128)

where REG-128 is an analogous ob (i.e. describes the frequency of a modulant) for the RF section's

-----  
<sup>1</sup> In actual fact the COMPLAINT stated that the amplitude was to be taken from either the '{f.101 \*}' or the '{l.101 \*}' generator. In setting up the test situation, however, the latter generator was eliminated from consideration, thereby excluding it from the generated CHOSEN-FROM commentary.

antenna input signal. In the last paragraph we saw how the value of that very ob was variabilized as 't'. WATSON remembers having done the variabilization. But the variabilization does not map straight away into a variabilization in the RF section's output signal because the mismatch at the output is due to harmonic distortion of the 1 kHz modulant. WATSON uses a new variable for the modulation frequency to be chosen from the generator of harmonics of t, hence

...  $w \in \{\sim t\}$  ...

To understand the variabilizations of the signal descriptors for AGC2 we must recall the way in which the output of the IF strip was explained in terms of its inputs, and how this explanation firmly affixed blame to AGC2. The IF strip, operating under normal circumstances is expected to have one carrier component -- carrier frequency, 455 kHz -- with one modulant -- modulation frequency, 1 kHz. The violation of the class A constraints for the IF strip resulted in the prediction of non-fundamental carrier and modulation harmonics in the output signal. Hence the DISTORTION descriptor (on the signal of AGC2's port called IF-DET) is provided by the forward reasoning process followed by successfully matching the newly predicted signal with the described signal. Similarly, the MODULATION descriptor comes from forward reasoning and SIG-MATCH's success. The variabilization of 'u' is similar to that of 'w' above. In this case the carrier frequency ob, REG-129, has commentary indicating that the source of its value is ultimately the IF frequency parameter of the RF section. Since this is set by the design, the CARRIER-FREQ descriptor is not variabilizable *a priori*. But WATSON is working in the context of prevailing harmonic distortion of the signal. Hence he can variabilize with respect to choices from the generator for such distortion, leading to the declaration

...  $u \in \{\sim \sim(\text{IF-FREQ pf})\}$  ...

The variabilization of 'x' is an artifact of the distortion situation as well. One can think of harmonic distortion as always being virtually present where the non-fundamental components have negligible amplitudes. The variabilization of 'x' is an emphasis of the fact that the

harmonics have been realized. Finally the modulation frequency descriptor variabilization is a product of the fact that the signal modulation was variabilized at the port, DET-OUT, via a RULE-IDENTIFIED-WITH chain that went through this ob. The declaration of generator for  $w$  could equally well have been done at this point, but it is WATSON's predilection to mention the generator declaration at the first occurrence of the variable whose value is to be generated.

The only variabilization occurring among the outputs of AGC2 appears in the amplitude descriptor. This variabilization too arises because of the forward reasoning with respect to the improperly used IF strip. Let REG-130 be the ob corresponding to the signal's modulation frequency descriptor. Recall that the caveat alarm was triggered by the lack of sufficient bias at the bias port of the IF strip. The subsequent matching process (invoking OP-SPEC-MATCH) revealed that the actual bias present at BIAS-PORT of AGC2 corresponds to class C operation of the IF strip. Having noticed this, WATSON leaves a comment on REG-130

(ID-BY-USE-WITH REG-130 REG-131)

where REG-131 is denoted by  $\sim!$ "(CLASS-C-BIAS IF-STRIP ,pf) in the present context. In abstracting the bug, WATSON sees this comment and looks at comments on REG-131 to see how the latter's value might be determined. It is ultimately set from a generator that is run at design assembly time. Hence WATSON variabilizes the amplitude descriptor, declaring it to be assignable via the re-running of the generator that would determine the of value of the CLASS-C-BIAS parameter at design assembly-time

...  $y \in \{ \approx \sim(\text{CLASS-C-BIAS IF-STRIP pf}) \}$  ... .

The concocting of the rest of the code body is straight-forward. As usual the predicate of the outer COND is a matching of observed signals against abstracted signals. If this match is successful, the COND constructed from the prevailing binding of reason is evaluated. In the localization from which the abstraction is being done, reason is a list of two SIG-MATCHes of observed signals with abstracted signals. These signals correspond to the inputs and outputs of

AGC2 which are inconsistent with its correct operation. The explanation of mechanism given by (MISSING-PURPOSES ... ) arises from WATSON's having asked if anyone was responsible for providing a bias at the port between the AGC2 and the IF strip. Indeed, the plan for the RF section indicates that one of the purposes of the AGC2 is to provide such a bias.

### 8.3 The front-end.

WATSON's successful discovery of the misaligned front-end in the GR-78 results in two bug abstractions. The first abstraction arises by noticing that the failure of the RF section to reject the image station is due to the misalignment of its filters. The second abstraction suggests that the appearance of a station at 910 kHz down from the selected station is due to the lack of image rejection in the RF amplifier. The first abstraction is formalized as

```
(BUG B-132 RF-AMP
  (SYMPTOMS pf
    (INPUTS (~(RF-IN pf) sig1))
    (OUTPUTS (~(RF-OUT pf) sig2)))
  (SIGNS part . sign)
  (MISSING-PURPOSES . mp)
  "AUX" (sig-133 sig-134
    (part1 INPUT-FILTER) (part2 OUTPUT-FILTER))
  (GEN-SIGNALS
    (RF-AMP pf
      (INPUTS
        (sig-133 = RF-IN
          (CARRIER-COMPONENTS
            ((CARRIER-FREQ r = <s - 2 * ~(IF-FREQUENCY pf)> € {*)
              (MODULATION-COMPONENTS
                ((MODULATION-FREQ t € {*)
                  ))))))))
        (OUTPUTS
          (sig-134 = RF-OUT
            (CARRIER-COMPONENTS
              ((MODULATION AM)
                (CARRIER-FREQ r)
                (MODULATION-COMPONENTS
                  ((MODULATION-FREQ t)
                    ))))))))
      (CONTROL-BINDINGS
        (TUNING s)))
    (COND ((AND (SIG-MATCH sig-133 sig1) (SIG-MATCH sig-134 sig2))
      (COND ((NOT (ALIGNED
        ~(BAND-PASS part1 pf)
        ~(BAND-PASS part2 pf)
        STAGGERED))
```

```

(NOTE
  !"BUG B-132 RF-SECT
    (SYMPTOMS ,pf
      (INPUTS (~(RF-IN ,pf) ,sig1))
      (OUTPUTS (~(RF-OUT ,pf) ,sig2)))
    (SIGNS ( ALIGNMENT ,part1 ,part2)
      (TEST
        (NOT (ALIGNED
          ~(BAND-PASS ,part1 ,pf)
          ~(BAND-PASS ,part2 ,pf))))))
    (MISSING-PURPOSES . mp = (
      ,part1
      (CARRIER-REJECTION ... )
      ,part2
      (CARRIER-REJECTION ... )))))))))).

```

The declaration of 'r' is the only variabilization that introduces a new facet of the generalization (by variabilization) problem. This declaration says that the carrier frequency descriptor can take on any value that is legal. That value is further specified to be the image of the "tuned" frequency. Again this declaration is concocted by a process of pointer-chasing.

The carrier frequency descriptor for the RF amplifier's input is identified with the analogous ob on the RF section's antenna port signal. This ob, REG-135, was commented by the triggering of the caveat alarm that noticed the lack of image rejection (see section 6.2). The commentary looks like

```
(CTL-OFFSET-WRT REG-135 ~(TUNING PF-19) (BY <2 * ~(IF-FREQ PF-19)>))
```

indicating that the frequency is the image frequency of that allegedly selected by the tuning control of the RF section. As in the abstraction of Q301, the pointer chasing continues in order to locate the source of the tuning parameter. The ob that ~(TUNING PF-19) denotes is commented as having taken its value by virtue of an existential choice in the setting up of the test case (recall the COMPLAINT of section 6.1), with the further specification that this not be the same value as that of the carrier frequency of the input signal. It is not exactly correct to say that these values are existentially specified. They were, in fact, formerly so specified. The code associated with the caveat structure also chased down these pointers we have been looking at, and commented the frequency ob (for tuning and carrier) as being arbitrarily specified but relatively constrained by

an offset. So in tracking down how the carrier frequency was determined in the present abstraction, WATSON first makes it arbitrarily assignable from legal values ( ...  $r \in \{*\}$  ... ) but then tracing down the tuning control binding, he finds that it is the same as  $r$ , but offset. So he modifies the declaration of  $r$  to be cognizant of the offset, hence

...  $r = \langle s - 2 * \sim(\text{IF-FREQ pf}) \rangle \in \{*\}$  ...

The abstraction code body is the usual signal matching followed by by a test that corresponds to the extant binding of reason. Recall that WATSON placed the blame on the filters of the RF amplifier by examining their "aligned-ness." This is reflected in the predicate clause of the inner COND which tests that property. If the filters are not aligned, B-122 is NOTed. Observe that the part sub-expression of the BUG item is a list rather than an atom. This is because the bug is due to the synergistic interaction of parts. The list enumerates the parts involved and states the nature of their interaction.

The second bug abstraction is done with respect to the next level up in the plan-fragment hierarchy of the design. This corresponds to the application of LOCAL to the plan-fragment that realizes the plan for the RF section. The formal statement of the abstraction is

```
(BUG B-136 RF-SECT
  (SYMPTOMS pf
    (INPUTS ( $\sim$ (ANT-IN pf) sig1))
    (OUTPUTS ( $\sim$ (DET-OUT pf) sig2)))
  (SIGNS part . sign)
  (MISSING-PURPOSES . mp)
  "AUX" (sig-137 sig-138 sig-139
    sig-140 (part  $\sim$ (AGC2 pf)))
  (GEN-SIGNALS
    (RF-SECT pf
      (INPUTS
        (sig-137 = ANT-IN
          (CARRIER-COMPONENTS
            ((MODULATION-COMPONENTS
              (CARRIER-FREQ  $r = [s - 2 * \sim(\text{IF-FREQ pf})] \in \{*\}$ )
              (MODULATION-COMPONENTS
                ((MODULATION-FREQ  $t \in \{*\}$ )))))))
        (OUTPUTS
          (sig-138 = DET-OUT
            (CARRIER-COMPONENTS
              ((MODULATION AM)
                (CARRIER-FREQ 0.))
```

```

      (MODULATION-COMPONENTS
        ((MODULATION-FREQ t € (*)))))))))
    (CONTROL-BINDINGS
      (TUNING s)))
  (RF-AMP part
    (INPUTS
      (sig-139 = ANT-IN
        (CARRIER-COMPONENTS
          ((MODULATION AM)
            (CARRIER-FREQ r)
            (MODULATION-COMPONENTS
              ((MODULATION-FREQ t)))))))
      (OUTPUTS
        (sig-140 = RF-CNV
          (CARRIER-COMPONENTS
            ((MODULATION AM)
              (CARRIER-FREQ r)
              (AMPLITUDE u € {> 0.})
              (MODULATION-COMPONENTS
                ((MODULATION-FREQ t)))))))
        (COND ((AND (SIG-MATCH sig-137 sig1) (SIG-MATCH sig-138 sig2))
          (COND ((AND
            (SIG-MATCH sig-139 ~(SIGNAL ANT-IN pf))
            (SIG-MATCH sig-140 ~(SIGNAL RF-CNV pf)))
            (NOTE
              !"BUG B-136 RF-SECT
                (SYMPTOMS ,pf
                  (INPUTS (~(ANT-IN ,pf) ,sig1))
                  (OUTPUTS (~(DET-OUT ,pf) ,sig2)))
                (SIGNS ,part
                  (INPUTS (~(ANT-IN ,pf) ,sig-139))
                  (OUTPUTS (~(RF-CNV ,pf) ,sig-140)))
                (MISSING-PURPOSES . mp = (
                  ,part
                  (CARRIER-REJECTION ... )))))))))).

```

The variabilizations in this abstraction are all analogous to those in abstractions already presented. The BUG's code body is concocted in precisely the same way as the code body in the abstraction of C422. The purposes in the (MISSING-PURPOSES ... ) commentary in both the bug abstractions of this section are like the purpose commentary revealed as the cause of the caveat alarm of section 6.2. The missing purposes are simply statements of the fact that the filters and the RF amplifier are failing in their job of image rejection. These facts are gleaned from the combination of the caveat alarm and subsequent examination of the purposes commentary on the parts of the RF amplifier.

## 8.4 Q407.

The successive localizations done in the final scenario, which eventually lead to pinning the blame on Q407 -- a transistor in the audio driver amplifier -- do not engender bug abstraction. Recalling the localization sequence, we note that all localizations (until arriving at the audio driver) result from the use of tracing. Localization thereafter is based on reasoning about CIRCUIT plans. In any case these localizations are not abstractable under the criteria imposed by WATSON.

## 8.5 Using an abstraction.

Since (relevant) previously abstracted bugs were not available during WATSON's execution of the scenarios, it seems appropriate to re-run a scenario with the aid of such an abstraction. We shall reconsider the scenario involving front-end misalignment. For the purposes of this illustration we shall ignore the existence of the MISSING-PURPOSES sub-expressions of the abstracted bugs since they are cumbersome and are not used here. Referring to section 6.2, we imagine that the trouble has already been localized to the RF section. WATSON (in the scope of LOCAL) evaluates

```
(FIND ALL (bugname plan part sign)
  !" (BUG bugname RF-SECT
    (SYMPTOMS PF-19
      (INPUTS (~(ANT-IN PF-19) ~(OBS-SIGNAL ANT-IN PF-19)))
      (OUTPUTS (~(DET-OUT PF-19) ~(OBS-SIGNAL ANT-IN PF-19))))
    (SIGNS part . sign)))
```

This matches the invocation pattern of the second BUG, B-136, abstracted in section 8.3. The code body of that BUG runs and NOTEs the item

```
(BUG B-136 RF-SECT
  (SYMPTOMS PF-19
    (INPUTS (PORT-141 SIG-142))
    (OUTPUTS (PORT-143 SIG-144)))
  (SIGNS PF-76
    (INPUTS (PORT-141 SIG-142)
      (OUTPUTS (PORT-145 SIG-146))))
```



which points at the plan-fragment PF-76 -- denoted by  $\sim$ (RF-AMP PF-19).

At this point the sign of the RF amplifier has already been examined (by the BUG's code body). So WATSON can proceed directly to locating the trouble inside. He asks again if there is a known bug (whose symptom is the same as the sign returned above). He does this by evaluating

```
(FIND ALL (bugname part sign)
!"(BUG bugname AMP2 ;AMP2 is the plan for the RF amplifier.
  (SYMPTOMS PF-76
    (INPUTS (PORT-143 SIG-144))
    (OUTPUTS (PORT-147 SIG-148)))
  (SIGNS part . sign)))
```

which invokes the abstracted BUG, B-132, of section 8.3. This leads to the NOTEing of

```
(BUG B-132 AMP2
  (SYMPTOMS PF-76
    (INPUTS (PORT-145 SIG-146))
    (OUTPUTS (PORT-149 SIG-150)))
  (SIGNS (ALIGNMENT PF-78 PF-77)
    (TEST (NOT (ALIGNED PARAM-151 PARAM-152))))).
```

The (ALIGNMENT ... ) tells WATSON that he need not pursue any further explanation of the bug.

## 9 Signals

In this chapter we shall examine the details of WATSON's manipulation of signals. Included among these details are the data structures for representing signals, the rules underlying the forward reasoning that models signal propagation, the procedures for matching two signals, and the measurement of signals in the receiver undergoing trouble-shooting. To make matters concrete, I shall draw from examples of signal manipulation necessitated by the execution of the scenario concerning the capacitor, C422, in chapter 5.

Before jumping into the details of signal manipulation, a few words are in order about the representation of signals. A signal is a data structure associated with a port in a particular plan-fragment. The same data structure will be accessible through ports in different plan-fragments if those ports meet at a port of their containing plan-fragment. A signal structure is a collection of obs standing in a hierarchical relationship like the one presented schematically in figure 9.1. Descending from the top we see a signal attached to a port. In turn there are obs below that say how the signal is electrically represented (e.g. voltage or current), tell what the DC bias is thought to be, and indicate the presence of various carrier components. Carrier components in turn have their own features like amplitude and carrier frequency. Indeed, this representation of a signal has much in common with the classical frequency domain representation of signals. The difference lies in that fact that all features of interest are presented explicitly in the data structure. For example, there is no way to label a component in a Fourier sum as having arisen due to distortion. Moreover, though a Fourier sum implicitly represents the power ratios of various components, the WATSON's signal representation may bear explicit commentary to the effect that components A and B are in the ratio of two to one

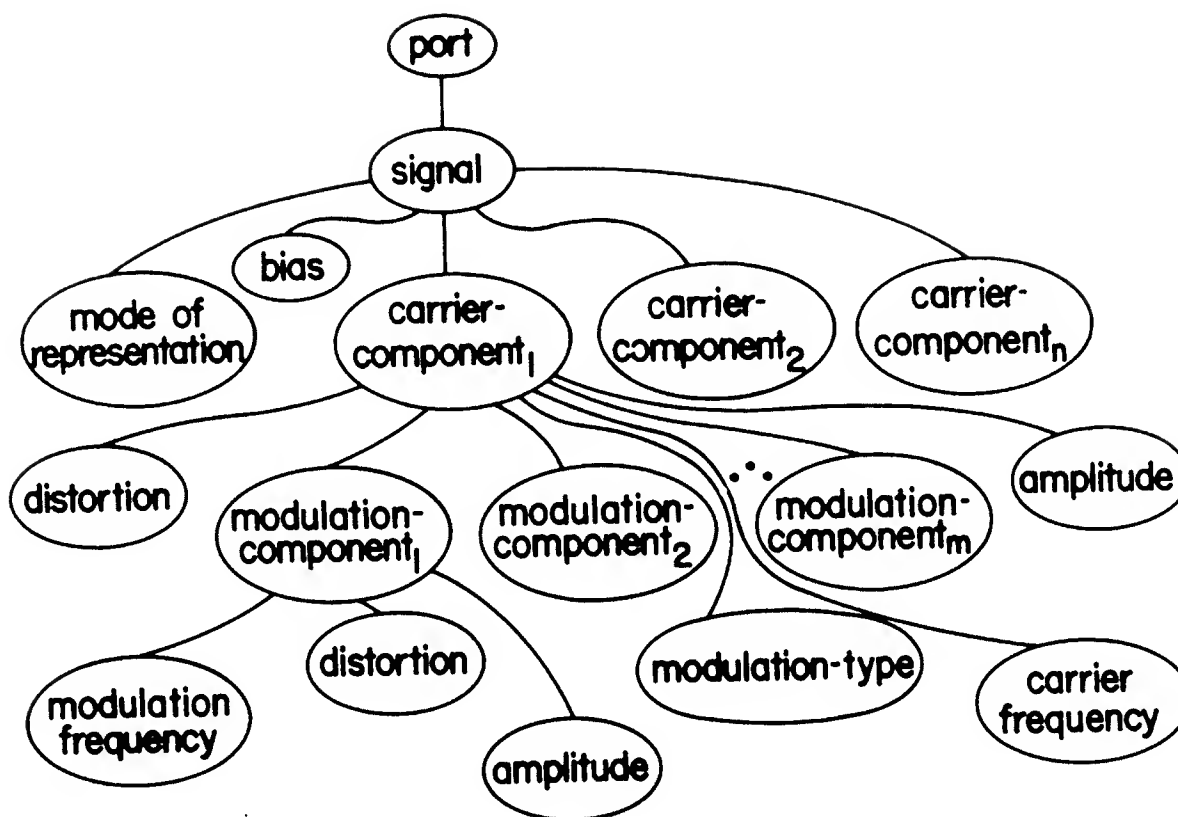


Figure 9.1 -- The ob hierarchy in the signal representation.

independently of what their amplitudes happen to be. Hence, a wide band amplifier might be described as one which preserved such ratios. The final point to be made about the representation is that though the obs which are the principal actors in a signal description may take on values, they need not do so. Thus the presence of a 455 kHz carrier may be specified without saying anything about its amplitude. A Fourier description in contrast puts all the information in the Fourier coefficients. This is the sense in which WATSON's signal descriptions are qualitative: much can be said about a signal without specifying it completely.

### 9.1 Matching and propagation on the top-level plan-fragment.

In the case of C422 WATSON's first order of business was to match the expected signal

at the GR-78's speaker output port against the observed signal<sup>1</sup> at that port. Let's suppose that the expected and observed signals are respectively SIG-153 and SIG-154. WATSON evaluates

```
(SIG-MATCH 'SIG-153 ,SIG-154).
```

After some initialization (like the creation of a new context to record matching commentary) SIG-MATCH calls SIG-MATCH1. (Refer to figure 9.2 for the flow of control in signal matching.) SIG-MATCH1 handles first layer descriptions of signal structure including (especially) the various carrier components. In particular SIG-MATCH1 tries to match SIG-153 against SIG-154 carrier component by carrier component. To that end SIG-MATCH1 must obtain the expected carrier components.

WATSON evaluates

```
(FIND ALL (cc) ' (CARRIER-COMPONENT SIG-153 cc)).
```

There is no item in the data base that matches the pattern in the FIND. There is, however, a matching IF-NEEDED method -- a determiner of the kind mentioned in section 4.1. This determiner looks like

```
(IF-NEEDED DET-155 (CARRIER-COMPONENT SIG-153 x)
  (FOR ALL (cc) ' (CARRIER-COMPONENT ~(SIGNAL ANT-IN PF-3) cc)
    (NOTICE ' (CARRIER-COMPONENT ~(SIGNAL ANT-IN PF-3) cc))))).
```

NOTICE is like ADD in that it triggers whatever IF-ADDEDs would have been triggered had an ADD of the argument pattern been done, but without the effect of putting the item in the data base. The reason for doing the NOTICE is to activate the IF-ADDEDs that formulate the rules governing the behavior of PF-3. Since the IF-ADDEDs take each of the carrier components through the receiver independently (the meaning of linearity), each of the carrier components coming into the top-level plan-fragment must be NOTICED separately. Such IF-ADDEDs are called the *agents* of a rule. The agent of interest here is

-----

<sup>1</sup> In section 4.1 I discussed matching the reported and expected signals. Since the matching involved there is essentially the same as matching expected and observed signals, I shall restrict our attention to this latter kind of match.

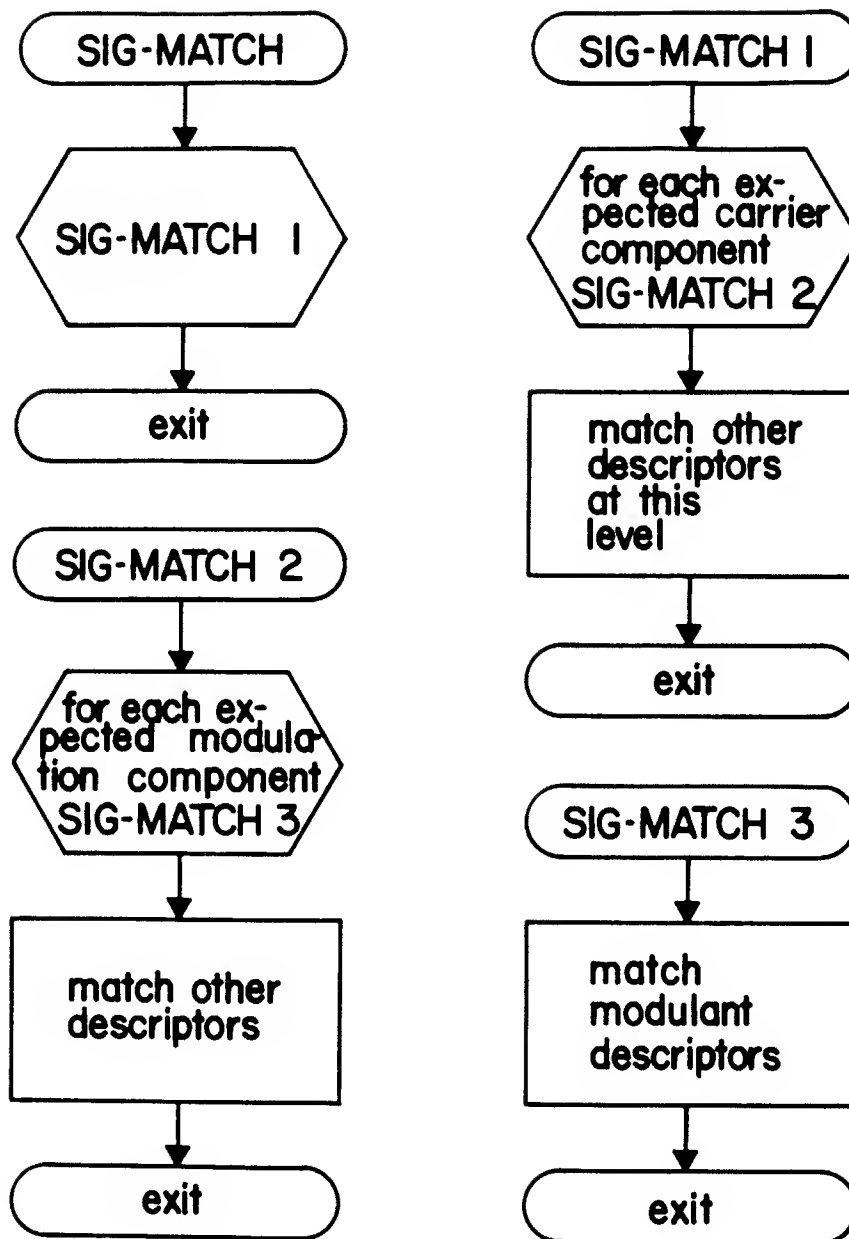


Figure 9.2 -- Basic control structure in SIG-MATCH.

```

(IF-ADDED AGENT-156 (CARRIER-COMPONENT ~(SIGNAL ANT-IN PF-3) cc)
(COND ((EQUAL
    .. ~(CARRIER-FREQ CARRIER-COMPONENT SIGNAL ANT-IN PF-3)
    .. ~(TUNING-FREQ PF-3))
(ADD 'DET-157 GR-78)
(ADD '(CARRIER-COMPONENT SIG-153 CC-158) GR-78)

```

```
(ADD !" (COMPONENT-CORRESPONDENCE CC-158 ,cc) GR-78)
(NOTE ' (CARRIER-COMPONENT SIG-153 CC-158))))).
```

It says that if a carrier component should appear at the antenna port of the top-level plan-fragment, and if that carrier has the "right" frequency (the frequency to which the receiver is tuned), then audio appears at the speaker port of the radio<sup>1</sup>. Notice that a method called DET-157 was also ADDED. This newly activated method is a determiner which will mediate the specification of the undetermined carrier properties. In any event the first FIND that was done eventually returns the item

```
(CARRIER-COMPONENT SIG-153 CC-158).
```

SIG-MATCH1 wants to match an expected signal against an observed signal. There is now in hand a single expected carrier of unknown properties. In order to advise MAXWELL what to look for in the way of observed signals, SIG-MATCH1 needs the carrier frequency(ies) of the expected carrier component(s). To get them, the usual

```
(FIND (THE-ONLY 1) (cf) ' (CARRIER-FREQ CC-158 cf))
```

is evaluated. As before the data base does not presently contain a CARRIER-FREQ item of the correct variety. To remedy this the determiner DET-157 is invoked. This determiner looks like

```
(IF-NEEDED DET-157 (CARRIER-FREQUENCY CC-158 cf1)
  (FOR (THE-ONLY 1) (cc) ' (COMPONENT-CORRESPONDENCE cc CC-158)
    (FOR (THE-ONLY 1) (cf) !" (CARRIER-FREQUENCY ,cc cf)
      (NOTICE !" (CARRIER-FREQUENCY ,cc ,cf))))))
```

which traces back along the carrier component identification provided by the COMPONENT-CORRESPONDENCE item and suggests taking note of the frequency descriptor hanging from the corresponding carrier component on the input to PF-3. This causes the agent

```
(IF-ADDED AGENT-159 (CARRIER-FREQUENCY cc cf)
  (FOR ALL (cc1) !" (COMPONENT-CORRESPONDENCE ,cc cc1)
    (FOR (THE-ONLY 1) (cf) !" (CARRIER-FREQUENCY ,cc cf)
      (ADD !" (CARRIER-FREQUENCY ,cc1 ,cf))
      (ADD !" (VALUE ,cf 0.))
    )
  )
)
```

<sup>1</sup> We shall shortly see that audio is a carrier component of 0 Hz carrier frequency, with modulation components of modulation frequencies less than 40 kHz.

```
(ADD 'OET-160)
(ADD 'AGENT-161)
(NOTE !" (CARRIER-FREQUENCY ,cc ,cf))))
```

to run. This results in the successful FINDing of a carrier frequency for the speaker output signal. Fetching a value for this frequency will yield 0. This connotes a DC carrier (a potentially audio signal) for WATSON.

Thus SIG-MATCH1 suggests to MAXWELL that the observed signal should be pure audio. On applying his frequency analyzer at the speaker port of the receiver, MAXWELL indeed measures no components beyond 5 kHz. In addition to the items

```
(PF-SIGNAL PF-3 PORT-2 SIG-153)
(PF-OBS-SIGNAL PF-3 PORT-2 SIG-154)
(CARRIER-COMPONENT SIG-153 CC-158)
(CARRIER-COMPONENT SIG-154 CC-162)
(CARRIER-FREQUENCY CC-158 CF-163)
(VALUE CF-163 0.).
```

MAXWELL ADDs the items

```
(MATCH-COMPONENT-CORRESPONDENCE CC-158 CC-162)
(CARRIER-FREQUENCY CC-162 CF-164)
(SIG-MATCH CF-163 CF-164)
(VALUE CF-164 0.)
```

asserting that the observed and expected carrier match as far as their frequency descriptors are concerned.

Now WATSON is in a position to be able to match the two signals at deeper levels of descriptor detail. SIG-MATCH1 calls SIG-MATCH2 for each carrier component in the expected signal. SIG-MATCH2 first asks what sort of modulation is expected on the CC-158 component of SIG-153. This results in an answer of 'AM' via the now familiar techniques of determiners and agents. SIG-MATCH2 proceeds to examine other descriptors of the carrier component under scrutiny<sup>1</sup>

-----  
<sup>1</sup> At every level of matching -- SIG-MATCH1, SIG-MATCH2, SIG-MATCH3 -- there are many candidate descriptors to be matched. Some descriptors -- like carrier amplitude descriptors, or descriptors giving the electrical representation -- would be inappropriate to ask about in the current plan-fragment. Indeed the match subroutines try to fetch all possible descriptors but in many cases they will get nothing for their efforts. Since matching is driven by the expected signal, the lack of a particular kind of descriptor will cause the matcher to lose interest in

starting with the modulation components.

As with the carrier components, WATSON gets the modulation components on the carrier CC-158 (Refer again to figure 9.2, especially the use of SIG-MATCH3.) by evaluating

```
(FIND ALL (mc) !" (MODULATION-COMPONENT CC-158 mc))
```

which finds no *a priori* present item, but does trigger the determiner

```
(IF-NEEDED DET-160 (MODULATION-COMPONENT CC-158 mc)
  (FOR (THE-ONLY 1) !" (COMPONENT-CORRESPONDENCE CC-158 cc1)
    (FOR ALL (mc) !" (MODULATION-COMPONENT ,cc1 mc)
      (NOTICE !" (MODULATION-COMPONENT ,cc1 ,mc))))
    (FOR ALL (mc1) !" (MODULATION-COMPONENT CC-158 mc1)
      (NOTE !" (MODULATION-COMPONENT CC-158 ,mc1))))).
```

This determiner in turn triggered a local agent placed on the alert by AGENT-156 above. This agent computes the audio modulants passed by the receiver and attaches them to the proper carrier component (in this case the DC carrier of frequency 0 Hz) at the output.

After the activity induced by the FIND of the last paragraph subsides, the data base has acquired the additional items

```
(MODULATION-COMPONENT CC-158 MC-165)
(COMPONENT-CORRESPONDENCE MC-166 MC-165),
```

which reports that the expected audio output mirrors the RF input modulation. SIG-MATCH2 now needs the frequencies of the various modulants (one in this case). An appropriate FIND will ADD to the data base the items

```
(MODULATION-FREQUENCY MC-165 MF-167)
(VALUE MF-167 1000.).
```

SIG-MATCH2 asks MAXWELL to find a similar modulant on the output port. When MAXWELL looks

-----  
matching it (and its sub-descriptors) to descriptors in the observed signal. Thus the "desire" to match is controlled by the *a priori* presence of certain classes of descriptors in the data base, or the presence of determiners to make up such descriptors. In any event, the "empty-handed" fetch is an indication that the descriptor being tried for is irrelevant to the localization at the prevailing level of detail.



for a modulant of a given frequency, he also tries to see if he can find harmonics of that frequency. MAXWELL reports to SIG-MATCH2 that there are 3 and 5 kHz modulation components in addition to the expected 1 kHz modulation. SIG-MATCH2 on hearing this news, ADDs to the data base the items

```
(MODULATION-COMPONENT CC-162 MC-168)
(MODULATION-COMPONENT CC-162 MC-169)
(MODULATION-COMPONENT CC-162 MC-170)
(MODULATION-FREQUENCY MC-168 MF-171)
(MODULATION-FREQUENCY MC-169 MF-172)
(MODULATION-FREQUENCY MC-170 MF-173)
(SIG-MATCH MC-168 MC-165)
(SIG-MISMATCH MC-169 MC-165)
(SIG-MISMATCH MC-170 MC-165).
```

SIG-MATCH2 calls SIG-MATCH3 for each carrier component of the expected signal that is relevant in the analysis of the top-level plan-fragment. Since there are none in this case no additional items are ADDed, other than a comment to the effect that harmonic distortion is present -- as is suggested by the modulation component mismatches. Having obtained a mismatch at the outputs (given correct inputs) WATSON looks inside the top-level plan-fragment<sup>1</sup>.

## 9.2 Propagation and matching in the RF section/AF section cascade.

Consider for a moment the plan of figure 2.4. Looking at it from the outside, WATSON sees three external interfaces -- the antenna, power and speaker ports. The first two of these are input ports, and their externality implies that the signals on them are completely under WATSON's control. (Recall the setting up of the test situations in the scenarios.) The signal at the output port is, of course, a function of the signals at the two input ports. On crossing the boundary of the top-level plan-fragment, the internal ports connecting the AF section and the RF section, the power supply and the RF section, and the power supply and AF section become visible. These

-----  
<sup>1</sup> Again, I must credit Winston's c-note structures as the inspiration for much of the relational machinery involved in the matching of expected and observed signals.

internal ports (like the speaker output port) have signals that are determined by the rules governing the parts of the plan(-fragment) together with the input signals crossing the boundary. We shall now see how a slight elaboration of the basic mechanisms introduced in the previous section can predictively account for the behavior of this cascade.

As before, the machinery is set in motion by requesting a match of the expected signal at the output of the AF section with the observed signal at that port. WATSON, via SIG-MATCH, tries, as before, to FIND the carriers embodied by the expected signal. This request wakes up a determiner which also evaluates a FIND, but for the purpose of obtaining the expected carrier components at the port coming from the RF section into the AF section. Let's call the expected version of this latter signal SIG-174. There is no item of the form

(CARRIER-COMPONENT SIG-174 ?)

in the data base. But there is a determiner for such a pattern. The determiner (an IF-NEEDED) tries in its turn to FIND the carrier components on the antenna input port to the RF section. This last port is identified with an external port of the top-level plan-fragment. That is, the antenna port seen outside the top-level plan-fragment, and the antenna-port coming into the RF section inside the top-level plan-fragment have the same port canonicalization, as indicated by PF-PORT items in the data base. There is a generally applicable IF-NEEDED for FINDing the descriptors of input signals at external ports. The gist of the IF-NEEDED's body has two parts: If the carriers of a signal at an external port are needed, then copy the descriptors of the signal at the port with the same canonicalization in the immediately superior plan-fragment. This might be sufficient except for the fact that it is possible to have plans like the one illustrated in figure 9.3. Suppose some signal detail is needed at the input port of D. The signal at the port between A and B may not be sufficiently detailed to supply the needed descriptor. Looking at C's output signal, however, may give the needed descriptor. So the second part of the IF-NEEDED concerns "looking across" to copy the needed descriptive details. The IF-NEEDED comments the copied components as having been "pulled down" from the superior plan-fragment.

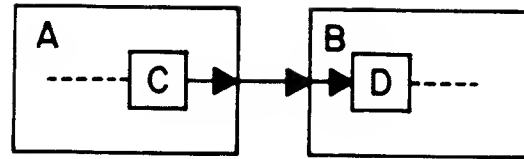


Figure 9.3 -- A port commonly shared at two levels.

When the FIND of the carriers at the output of the AF section successfully returns, the data base shows expected signals at the antenna input of the RF section with a single carrier of 5 MHz, between the RF and AF sections having a single carrier at 0 Hz, and at the audio output of the AF section having a single carrier of 0 Hz. These carriers are mutually identified by CARRIER-CORRESPONDENCE comments so that modulant (and other) descriptors may be properly appended<sup>1</sup>. At this point SIG-MATCH1 is holding on to all (one of) the expected carrier components of the AF section's output signal. As before, SIG-MATCH2 is invoked for each of these components. This reveals once again that the modulants in the expected and observed output signals are mismatched due to harmonic distortion, leading to mismatch commentary in the data base analogous (not surprisingly) to that shown in the last section for the speaker output port of the top-level plan-fragment (seen as a part). SIG-MATCHes are now done on the input signals of the AF section. Getting the expected input signal from the power supply (to the AF section) causes the triggering of determiners and agents (which were previously dormant) to compute the

-----  
<sup>1</sup> Some clarification is in order regarding the manipulation of plan-fragments for the purposes of signal matching. When starting the localization process, WATSON is really looking at the top-level plan-fragment as being the only part of an implied plan-fragment. The ports of this part happen to be equivalent under canonicalization to the external ports of the top-level plan-fragment. The inputs at these external ports (of the part) are known to be correct since they are under WATSON's control. SIG-MATCHing done in the complaint validation phase, is isomorphic to the usual input/output analysis that may result in a recursive application of LOCAL to a plan-fragment. In the present case the application of LOCAL pops into view the plan structure of figure 2.4.

description of the DC signal that should appear at the power port of the AF section. This expected signal straightforwardly matches the observed signal. SIG-MATCHing on the signal at the port connecting the RF and AF sections, however, reveals a mismatch isomorphic to that seen at the output of the AF section -- the presence of audio harmonics of the expected 1 kHz fundamental modulant.

Before looking elsewhere for the problem LOCAL wants to be sure that the observed output of the AF amplifier is reasonable in terms of its faulty inputs. The CONNIVER context machinery presents a handy mechanism for doing this rationalization. Recall that LOCAL, on entering the top-level plan-fragment, created a new context -- called c1 -- in which to do deductions. Having created that new context, LOCAL ADDED a number of methods to that data base (in that context) to facilitate thinking about signal propagation. All data base items created thereafter were stored either in this context or a sub-context created by SIG-MATCH. To do the rationalization, LOCAL creates a new context -- called c2 -- in parallel to c1, and simply activates in c2 all the methods activated in c1. LOCAL now reads the observed signal descriptors for the inputs of the AF section plan-fragment, and asserts them in c2 as if they were expected signal descriptors. This fires a bevy of methods and eventually results in a complete description at the output of the AF section. The latter signal is then SIG-MATCHed against the observed signal, yielding a successful match. Hence the output of the AF section is reasonable in terms of its inputs. LOCAL then moves its attention to the RF section, notices that its inputs are okay, and therefore applies itself recursively to the plan-fragment of the RF section.

Before moving on to the analysis of SIG-MATCHing expected and observed signals inside PF-19 -- the plan-fragment for the RF section, let's recapitulate what we have seen of signals thus far. LOCAL attempts to SIG-MATCH the observed outputs of a part against expected outputs. SIG-MATCHing requires FINDing the carrier descriptor items for the expected signal, for the latter are used as templates to drive the measurements process. Mostly such items will not be present, so they must be determined by appropriate IF-NEEDEDs. Typically such an IF-NEEDED

will FIND carrier components of the input signal(s) of the part under scrutiny. Of course these latter component items might not be around either, which leads to another level of determiner/agent interaction. Sometimes components will be gotten by looking up or across the tree of plan-fragments (the external input port case). In any event SIG-MATCH eventually obtains all of the components of the expected signal and instructs MAXWELL to find components of the same fundamental and harmonic carrier frequencies in the observed signal. After all the sub-procedures of SIG-MATCH finish, control is returned to LOCAL with a "match" or "mis-match" message, and a context containing the match correspondences (or differences). A successful match has occurred whenever the descriptors are entirely in correspondence. Mis-match can be brought about by disagreement of values for descriptors, or by the presence of extraneous descriptors such as harmonics.

### 9.3 Propagation and matching in the RF section.

Applying LOCAL to PF-19 leads to signal back-tracing the structure of figure 2.5. Signal mis-matches are discovered at the detector output and input ports. The two mis-matches are analogous. The expected signal at the IF strip output has only a 455 kHz carrier. MAXWELL reports a 910 kHz carrier as well. LOCAL requests the SIG-MATCHing of the inputs to the IF strip against the expected inputs. The signal at the converter/IF strip interface matches. The attempt to SIG-MATCH the expected and observed signals between the AGC2 and the IF strip triggers the caveat alarm discussed in section 5.2. We shall now investigate the interaction of the caveat machinery with WATSON's signal processing reasoning.

It should be evident by now that there are many IF-ADDED methods associated with modelling the behavior of parts in a plan-fragment. Note that these methods do not become active (PRESENT in the CONNIVER sense) until LOCAL is applied to the plan-fragment in which the part in question becomes visible. In other words, on applying LOCAL to a new plan-fragment, part

of the initialization of the new level of localization is the activation of all the agents, determiners, and other methods that become relevant at the newly entered level of detail. One such method activated is a caveat trigger that looks like

```
(IF-ADDED
  C-ALARMIST-175
  (VALUE
    ..~(AMPLITUDE CARRIER-COMPONENT OBS-SIGNAL IF-BIAS-PORT PF-19)
    ampl)
  (COND ((< ampl .6) ... ))).
```

This code was compiled at design assembly time. (See section 10.6 on caveat alarmist compilation). The body of the IF-ADDED is run only if the amplitude of the DC signal between the AGC2 and the IF strip is less than 0.6 V. The body of the method is not important in its details<sup>1</sup>. The essential feature is that it records in the frame of the nearest enclosing invocation of LOCAL the current program counter (in the CONNIVER sense) and a message describing the nature of the caveat that has been violated. Control is then passed to that frame at the point denoted by the tag :CAVEAT-ENTRY.

The way C-ALARMIST-175 is set off is through the SIG-MATCH done at the interface of the IF strip and AGC2. The story goes like this: SIG-MATCH tells MAXWELL to look for a carrier component of 0 Hz frequency at the AGC bias port of the IF strip. As usual, MAXWELL tries to glean all of the descriptors associated with that component. One of the things he discovers is that the signal has a DC voltage of 0 V, an important observation since this is a voltage port<sup>2</sup>. Thus he ADDs the item

```
(VALUE ~ (AMPLITUDE CARRIER-COMPONENT OBS-SIGNAL IF-BIAS-PORT PF-19) 0.)
```

to the data base, triggering C-ALARMIST-175. Control passes to the nearest instance of LOCAL, as

-----

<sup>1</sup> The invocation pattern of this method is also wrong in detail -- but correct in spirit. A more accurate version is given in section 10.6.

<sup>2</sup> MAXWELL gets this last piece of information by descending to more detailed levels of the plan-fragment hierarchy. Such "looking inside" is necessitated by the exigencies of measurement, i.e. whether a voltage measuring instrument or a current measuring instrument is appropriate.

previously stated. LOCAL inspects the message and notices that it was generated in the course of a SIG-MATCH<sup>1</sup>. LOCAL decides to let the SIG-MATCH proceed to completion and notes privately that a closer look at the IF strip should later be taken since it has an input that violates a caveat and a misbehaving output as well. After the SIG-MATCHing is completed for all of the IF strip's ports, control again reverts to LOCAL.

LOCAL now turns its attention in earnest to the caveat message placed on its agenda<sup>2</sup>. The AGC input to the IF strip is evidently not correct, which means that the IF strip's peculiar output may actually be reasonable. Verifying this reasonableness reveals another facet of WATSON's signal sagacity. As pointed out in section 5.2, WATSON seeks guidance in forward reasoning by looking for the source of the caveat. Having found the source, he flattens the plan-fragment for the IF strip modulo that source -- the plan-fragment for the IF1 amplifier. Schematically this means that figure 2.5 is transformed into figure 9.4. This flattening is necessitated by the observation that the violation of the caveat belies some assumption underlying the validity of the rule associated with the IF strip. By looking inside the IF strip, its behavior can perhaps be reformulated in the unexpected situation. The OP-SPEC-MATCHing mentioned in section 5.2 and discussed in detail in section 10.7 eventually results in activating a new rule for IF1 in the context of the 0 V bias from AGC2. The key observation to be made is that the flattened IF strip, with the new rule for IF1, can correctly account for the behavior of the overall IF strip<sup>3</sup>.

-----  
<sup>1</sup> Remember that LOCAL has been given the stack frame of the caveat trigger as part of the caveat message. The predecessors of this frame are amenable to examination so as to verify that a SIG-MATCH was in progress.

<sup>2</sup> The caveat catching machinery described does not provide for the mediation of many caveats in the scope of a given activation of LOCAL. Such caveat "interrupts" must be caught and fully "dismissed" before another is acceptable. If another should come along while one is "in progress" WATSON is considered to be in error. Of course, a facility for handling multiple simultaneous caveats could easily be constructed. The cost of maintaining a "forked" history of caveat frames seems great in comparison to the likelihood of needing such a hairy control strategy.

<sup>3</sup> Of course, having flattened things out, the signal propagation rules for the parts of the IF strip may show considerably more signal processing detail than the original propagation rule of the IF strip plan-fragment. This is only fair since WATSON had to appeal to more detailed plan information to determine what would happen to the IF strip in a prohibited situation.

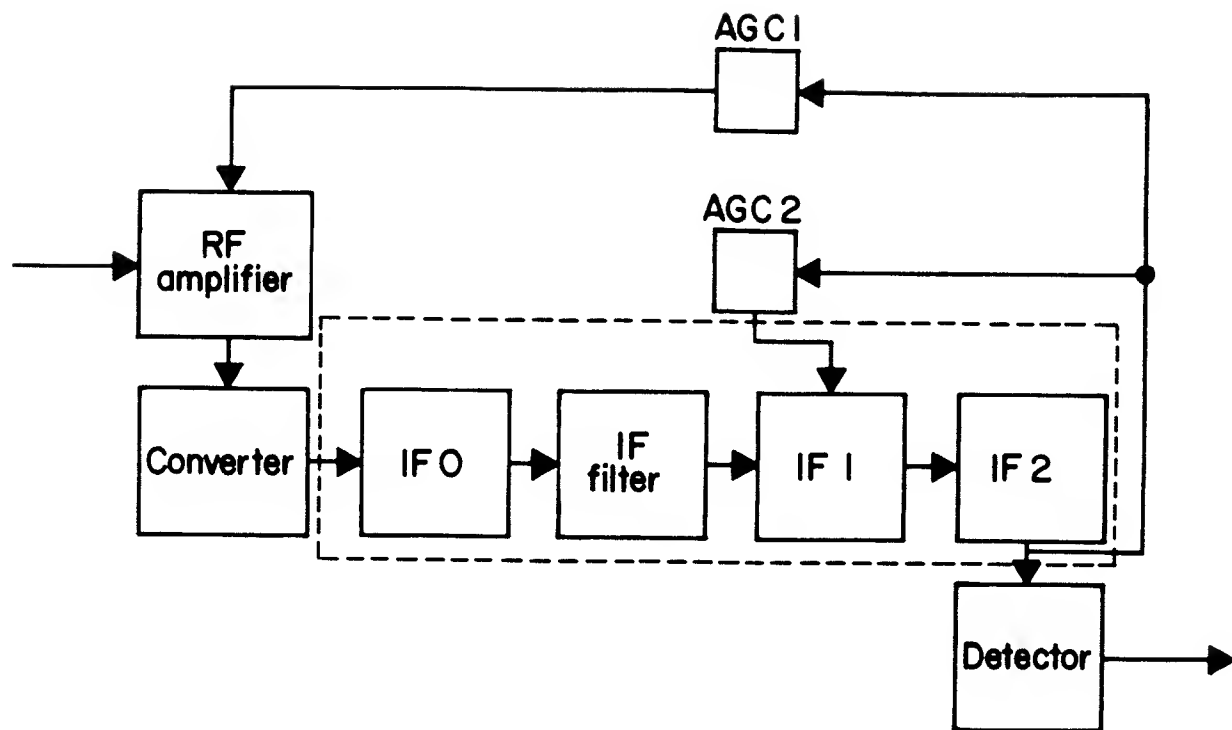


Figure 9.4 -- Flattening of the RF section modulo IF1.

Having created a presumably valid rule governing the behavior of the IF1 amplifier (and by cascading, the expanded IF strip) WATSON is ready to attempt rationalizing the output of the IF strip. The key activity occurs around IF1. (We will ignore the propagation of the expected signal through the other parts of the IF strip.) Let's imagine that WATSON is SIG-MATCHing at the output port of IF1 and has managed to arrive at this port without having previously constructed any of the predicted signal structure for the port<sup>1</sup>. The SIG-MATCH of the predicted signal structure, SIG-176 against the observed signal structure, SIG-177, leads to the evaluation of the usual

```
(FIND ALL (cc) ' (CARRIER-COMPONENT SIG-176 cc)).
```

The determiner for this descriptor is the now familiar piece of code

<sup>1</sup> This cannot be the actual situation because back-tracing through IF2 would have caused the generation of such a signal structure. The predicted signal structure generation that we are now looking at would have happened already.



```
(IF-NEEDED DET-178 (CARRIER-COMPONENT SIG-176 x)
  (FOR ALL (cc) ' (CARRIER-COMPONENT SIG-179 cc)
    (NOTICE !" (CARRIER-COMPONENT SIG-179 ,cc)))
  (FOR ALL-ITEMS (cc) (CARRIER-COMPONENT SIG-176 ,cc)
    (NOTICE !" (CARRIER-COMPONENT SIG-176 ,cc))))),
```

where SIG-179 is the signal at the RF input port of the plan fragment for IFI. As usual the rule governing IFI has an agent listening for NOTICES from DET-178. The agent in question leaves in the data base a number of new items:

```
(CARRIER-COMPONENT SIG-176 CC-180)
(CARRIER-FREQUENCY CC-180 CF-181)
(VALUE CF-181 455000.)
(OISTORTION SIG-176 EVEN-HARMONIC)
(MODULATION-COMPONENT CC-180 MC-182)
(MODULATION-FREQUENCY MC-182 MF-183)
(VALUE MF-183 1000.)
(OISTORTION MC-182 EVEN-HARMONIC)
```

and a method

```
(IF-NEEDED OISTORT-MODULANT-184 (OISTORTION mc d)
  (FOR (THE-ONLY 1) (cc) !" (MODULATION-COMPONENT cc ,c)
    (FOR (THE-ONLY 1) (sig) !" (CARRIER-COMPONENT sig ,cc)
      (COND ((EQ sig 'SIG-176)
        (NOTE !" (OISTORTION ,mc EVEN-HARMONIC)))))).
```

Before the significance of all this can be made clear, a little more must be said about SIG-MATCH's interaction with MAXWELL. I mentioned earlier that when MAXWELL is told to go and find a particular carrier component, he generally looks for nearby harmonics of the requested component as well. In other words, SIG-MATCH tries to suggest a mis-match by finding harmonic distortion in the signal. Now this sort of behavior should be modifiable when harmonic distortion is expected. Both SIG-MATCH1 and SIG-MATCH3 look for distortion commentary on the components (carrier and modulation) to which they are applied. There is an explicit comment of that variety on the carrier CF-181 above. When SIG-MATCH2 receives a report from MAXWELL of harmonics of the 455 kHz fundamental, that is taken as being perfectly proper and it is a cause for creating more carrier components for the expected signal.

Similarly, when SIG-MATCH3 receives reports from MAXWELL of harmonics of the

modulation, there is no cause for alarm. SIG-MATCH3 simply creates appropriate new modulants to be attached to CF-181. The need for the IF-NEEDED arises because the output of IF1 should be expected to have modulation distortion for all the harmonics of the carrier that appear. The trouble is that there may not yet be appropriate carrier component obs to attach modulation distortion descriptors to. When SIG-MATCH3 reaches for a possible distortion commentary on a modulant, a procedure is activated to check whether the modulant in question is descended from the expected output signal of IF1. If so, distortion commentary is NOTed and things proceed just as before.

As a result of these items and methods having been ADDED to the data base, SIG-MATCH successfully matches the expected output signal with the observed output signal for IF1. Specifically, harmonic distortion of both carrier and modulation is to be expected. This successfully propagates through IF2 to the output of the IF strip (the output of IF2), hence accounting for the bad output of the IF strip in terms of its good input from the RF amplifier and the bad input from the AGC.

#### 9.4 Signals as voltages and currents.

One more aspect of signal description and propagation needs clarification in order to complete the picture of WATSON's handling of signals. We shall presently look into his methods of shifting from the abstract signal viewpoint to the voltage/current viewpoint. I shall continue to draw from the scenario of the AGC bug. WATSON, having satisfied himself that the output of the IF strip is reasonable, turns his attention to the problem localizing the bug further. Determiners and agents once again conspire in back-tracing to show that the AGC2's output is incorrect in light of its input from the IF strip. So LOCAL is applied to the plan-fragment of the AGC2. The heuristic of analogous signals at the input and output of a coupling plan immediately makes the RC network of figure 2.12 suspect. Hence the plan-fragment for this network is opened up and C422 becomes the prime candidate for taking the blame.

The question I wish to answer here is "What is the source of the test voltages of section 5.3 that led to C422's indictment?" At the input port for the plan-fragment PF-67 there is a description of a signal that consists of a number of rectified carrier components, each having a carrier frequency that is a harmonic of 455 kHz, and each carrier component having various even harmonics of 1 kHz as modulators<sup>1</sup>. This signal is conceptual only. It is not observable, but is conceptually useful for the purposes of modelling the functional behavior of the AGC2 plan. The voltages described at the input port of the coupling plan for the RC network were stated to have been rising and falling at particular rates. These rates are computed from the frequencies -- carrier and modulator -- seen on the input signal. WATSON's model is that the carrier and modulation are to be analyzed separately as a succession of rising and falling ramps -- a "saw

tooth." The amplitude and frequency of the saw tooth is the same as that of the carrier or modulation, whichever is appropriate. From this information the rise(fall) rate of the saw-tooth is computable. Then WATSON ADDs an item of the form

```
(DC-BV
 (BETWEEN node1 node2)
 description-of-value
 (AT-RATE rate-value) )
```

The agents for R451 respond with an assertion concerning node B's (refer to figure 2.12) changing at a similar rate. Finally C422 (if it were operational) would remove the latter assertions if they claimed node voltage changes exceeding a certain rate. Otherwise the assertions would be allowed to stand. The saw teeth getting through from node A to node B are then re-assembled into abstract signal descriptions. As it turns out, the rules for this particular network let through carrier components of 20 Hz or less. A shorted C422 is a variation on the same theme. Its rules simply forestall any data base items' asserting changes at node B with respect to ground. Hence no saw tooth can get through. Hence the output signal is quiescent at ground.

-----  
<sup>1</sup> This expected signal description results from a straightforward propagation. The expected output signal computed (as we saw in the previous section) by the IF strip rule is pushed through the rectifying AVC amplifier.

## 10 Plans, Plan-fragments, and Designs

In the past several chapters I have attempted to show how WATSON makes use of a well documented design to focus in on the ultimate etiology of failure in an apparently malfunctioning radio receiver. How is such a documented design created? How are plan-fragments assembled into a design? How are plans instantiated in plan-fragments? How are plans specified? How are various active structures -- like caveats and rules -- specified and intantiated? These are some of the many questions about documentation that will be answered in this chapter. The method of investigating these questions will be to show the assembly process for a fragment of the GR-78 receiver. I shall specify a sufficiently large sub-tree of the GR-78 design (a tree of plan-fragments) so as to give answers to (and insights into) the questions just posed. The sub-tree in question is rooted at the top-level plan fragment and descends to the level of detail of the plan-fragment describing the ouput filter of AGC2.

### 10.1 Assembling the GR-78 design.

The use of the word 'assembly' to denote the process of specifying a receiver design is a deliberate attempt to bring to mind the well known program specification process of the same name. An important feature of the design assembler is its ability to delay action on various aspects of the design which have not yet been fully specified. To wit:

(DESIGN GR-78 (INSTANCE RECEIVER))

instructs the assembler to open a new design called GR-78 and instantiate the plan called RECEIVER as its top-level plan-fragment. As we shall see shortly, the instantiation of the

RECEIVER plan as a plan-fragment will not complete the design; rather, it will introduce additional aspects of the design to be specified. The specification of the design is terminated (as distinct from completed) when

(ENDESIGN GR-78)

is scanned by the assembler. If there should remain any loose ends -- unspecified aspects of the design -- at the time of closing the design, WATSON will complain.

The plan -- potentially usable in designs other than the GR-78 -- for RECEIVER is specified<sup>1</sup> as follows:

```
(DEFPLAN RECEIVER CASCADE
  (PORTS (INPUT (ANT-IN INFO) (WALL-AC POWER)) (OUTPUT (SPKR-OUT INFO))
    (INTERNAL (PWR POWER) (RF-AF INFO)))
  (CONTROLS TUNING VOLUME)
  (PARTS RF-SECT AF-SECT PWR-SPPLY)
  (GOALS . s-185)
  (RULE s-186)
  (CASCADE
    (RF-SECT (ANT-IN (INPUT PWR) (OUTPUT RF-AF)) s-187)
    (AF-SECT ((INPUT (RF-AF PWR) SPKR-OUT) s-188 )
      (PWR-SPPLY (WALL-AC (OUTPUT PWR)) s-189 ) ... )).
```

A plan called RECEIVER is declared to be of type CASCADE. It has a number of ports. An input port -- ANT-IN -- carries an information signal, while another port -- WALL-AC -- carries a power signal. An output port -- SPKR-OUT -- is also defined as carrying an information signal. There is a class of *internal* ports defined. These ports will be the interfaces among the various parts defined within this cascade plan<sup>2</sup>. Next we see a declaration of external controls for this plan. Also indicated are some goals -- declarative explanations of what the plan is to accomplish -- and a rule -- an imperative equivalent of the goals. Finally a number of parts -- RF-SECT, AF-SECT, and PWR-SPPLY -- are declared to be connected in a cascade through their common ports. Since the connecting ports are internal they must be further specified as to their directionality.

<sup>1</sup> Apart from the usual ellipsis construction indicating left-out parts, there may also be named ellipses of the form 's-185' indicating a sub-expression to be filled in later.

<sup>2</sup> It is possible for an *external* port to be an interface too. Such an example occurs in the OSC plan seen in chapter 2.

So far the RECEIVER plan shows the parts of the plan, how they are mutually connected, and how the plan is interfaced to the outside world (through input/output ports and controls). We shall defer, for now, filling out the declaration of goals for this plan, and consider the instantiation of the plan as a plan fragment. The evaluation of DESIGN above caused this instantiation to be carried out. Instantiation (in this case) simply reads the body of the DEFPLAN and compiles data base items precisely like those introduced in chapter 4, namely

```
(PF PF-3 RECEIVER GR-78)
(PF-PORT PF-3 ANT-IN PORT-1)
(PF-PORT-SANDWICH PF-3 PF-19 PORT-190 PF-20),
```

and so forth. At this point there is, for example, no definition for the part called RF-SECT. In order to guarantee that such a definition appears before the design is closed, an IF-ADDED

```
(IF-ADDED FWDREF-191 (PF-PART PF-3 RF-SECT part)
  (DELETE 'FWDREF-191 fwdrefs)
  (REMOVE 'FWDREF-191 'GR-78))
```

is salted away in the data base to await the eventual realization of a plan-fragment corresponding to the RF-SECT part in PF-3. The names of all such IF-ADDEDs are placed on a list called fwdrefs, which, if non-empty when the design is closed, precipitates an error. RF-SECT would actually be so realized by evaluating

```
(INstantiate (RF-SECT PF-3) SUPER-HET-RF)
```

where SUPER-HET-RF is a plan to be instantiated. Among other things, this instantiation will produce the item

```
(PF-PART PF-3 RF-SECT PF-19).
```

This establishes the interpretation of the part RF-SECT in PF-3 as PF-19. It should now be clear how the structure of parts and connections in a plan-fragment become established. One more remark about this simple passive structure is in order. We have frequently used path names like

```
~(SIGNAL ANT-IN PF-3).
```

This reference scheme is facilitated by the manufacture of a hierarchy of IS-A-IN relations manufactured during the design. In particular the following

```
(IS-A-IN PF-3 TOPLEVEL GR-78)
(IS-A-IN PORT-1 ANT-IN PF-3)
(IS-A-IN SIG-6 SIGNAL PORT-1)
```

yield the referenced ob, SIG-6.

## 10.2 Goals, purposes, and rules.

We are now going to fill in some of those interesting blanks left in the plan defined above. First, (GOALS . s-185) is really

```
(GOALS
  ((SELECT-FROM
    ,~(SIGNAL ANT ★)
    (CONTROLLED-BY TUNING))
    (PURPOSES ... ))
  ((DEMODULATE
    ,~(SIGNAL ANT-IN ★)
    ,~(SIGNAL SPKR-OUT ★))
    (PURPOSES
      (RF-SECT
        (DEMODULATE ,~(SIGNAL ANT-IN ★) ,~(SIGNAL RF-AF ★)
        (AF-SECT
          (MAKE (LINEAR-COPY ,~(SIGNAL RF-AF ★) ,~(SIGNAL SPKR-OUT ★))))))
      ((MAKE (AUDIBLE ,~(SIGNAL SPKR-OUT ★)))
        (PURPOSES
          (AF-SECT
            (MAKE
              (>> ,~(AMPLITUDE MODULANT CARRIER SIGNAL SPKR-OUT ★)
                ,~(AMPLITUDE MODULANT CARRIER SIGNAL RF-AF)))))))).
```

In plain English this says that the RECEIVER plan is trying to do three jobs -- select from the incoming signal, demodulate the incoming signal (putting out a suitable audio signal at the speaker), and make the output sufficiently loud to be heard. The RF-SECT and AF-SECT parts meanwhile are declared to have purposes, the fulfilling of which facilitates the achievement of

one or the other of the goals of the plan<sup>1</sup>. On realizing the plan as a plan-fragment, the goals will be compiled into

```
(PF-GOAL PF-3 RULE-192
  (DEMODULATE ,~(SIGNAL ANT-IN PF-3)
    ,~(SIGNAL SPKR-OUT PF-3))
  (~(RF-SECT PF-3) ~(AF-SECT PF-3)))
(PF-PURPOSE ~(RF-SECT PF-3) PF-3 RULE-193
  (DEMODULATE ,~(SIGNAL ANT-IN PF-3) ,~(SIGNAL RF-AF PF-3)))
(PF-PURPOSE ~(AF-SECT PF-3) PF-3 RULE-194
  (MAKE
    (LINEAR-COPY
      ,~(SIGNAL RF-AF PF-3) ,~(SIGNAL SPKR-OUT PF-3))))
```

and

```
(PF-GOAL PF-3 RULE-192
  (MAKE (AUDIBLE ,~(SIGNAL SPKR-OUT PF-3)))
  (~(AF-SECT)))
(PF-PURPOSE ~(AF-SECT PF-3) PF-3 RULE-194
  (MAKE
    (>> ,~(AMPLITUDE MODULANT CARRIER SIGNAL SPKR-OUT PF-3)
      ,~(AMPLITUDE MODULANT CARRIER SIGNAL RF-AF PF-3))))
```

which we observe are like the statements of plan-fragment goals which WATSON has found so helpful. RULE-194 is an instance of the rule governing the behavior of the RECEIVER plan, a rule that has not yet been exhibited. Also each purpose assigned to a part will eventually match a goal of the plan-fragment that realizes that part. The match is guaranteed by the same kind of forward referencing mechanism used to guarantee the integrity of the design *vis à vis* fully defining its parts.

Now we can look into the really interesting feature of a plan -- its rule. (RULE s-186) fully written is

```
(RULE
  (TO-GET !" (CARRIER-COMPONENT ,~(SIGNAL SPKR-OUT ★) ccl)
    (FOR ALL (cc) !" (CARRIER-COMPONENT ~(SIGNAL ANT-IN ★) ccl)
      (NOTICE !" (CARRIER-COMPONENT ,~(SIGNAL ANT-IN ★) ,cc1))))
  (ENTAIL !" (CARRIER-COMPONENT ,~(SIGNAL ANT-IN ★) ccl)
    (COND ((EQUAL .,~(CARRIER-FREQ CARRIER-COMPONENT SIGNAL ANT-IN ★)
      -----
```

<sup>1</sup> The descriptions of goals and purposes contain a gritch character, '★'. This is a place holder, which at the time of instantiation will refer to the plan-fragment produced.



```

      ..~(TUNING ★))
(ADD s-195 design-context)
(ADD
  !" (CARRIER-COMPONENT
    ,~(SIGNAL SPKR-OUT ★)
    , (CSETQ cc1 (GENSYM 'CC)))
  design-context)
(ADD !" (COMPONENT-CORRESPONDENCE ,cc1 ,cc) design-context)
(NOTE !" (CARRIER-COMPONENT ,~(SIGNAL SPKR-OUT ★) ,cc1)))))).

```

TO-GET expressions indicate how an output signal descriptor may be determined by input signal descriptors. ENTAILments describe what outputs the eventual plan-fragment should compute given certain inputs. The ENTAILment states that if the existence of a certain input carrier is asserted, and if this carrier is at the correct frequency, a certain output carrier is asserted. Though only one input is shown in this example, an ENTAILment may have any combination of disjuncts and conjuncts to trigger the running of the computational body.

### 10.3 Compiling determiners and agents.

The rule described in the last section needs to be transformed into a form appropriate to primitive CONNIVER constructs. The transformation of the TO-GET is straight forward. The relationship between the TO-GET and the determiner DET-155 of section 9.1 should be obvious. ENTAILments are another matter. The trouble is that in the general case, the running of the code body of the ENTAILment requires a collection of items to have been ADDED to the data base. This problem is not evident in the ENTAILment above since the code body is triggered by the appearance of a single item. CONNIVER IF-ADDED methods unfortunately are triggered by only one item. The rest of this section will be devoted to the technical details of compiling agents from ENTAILments.

The first order of business is to get the invoking pattern of the ENTAILment into a canonical form. Being a combination of conjuncts and disjuncts, a convenient canonicalization is provided by disjunctive normal form. The next thing done is to surround the code body by

```
(CLOSURE ' (CLAMBOA () code-body fr)
```

where *fr* evaluates to the CONNIVER frame in which the plan-fragment is being realized. Taking this closure of the code body insures that all the variables free with respect to the body of the ENTAILment will take on their proper values when the body is put to use<sup>1</sup>. Before continuing, let me point out that the basic strategy of agents will be to run the closure just computed whenever all of the conjuncts of any one of the disjuncts (in the canonical form) become known.

To carry out this strategy the agents will employ Moore's tactic [Moore, 1975] of "working from the left." WATSON makes a first-order agent for each disjunct. Such agents are IF-ADDEDs whose invoking patterns are the first conjuncts of each of the disjuncts for which the agents were respectively created. When the method is invoked by the ADDing of an item (matching the first conjunct), the method's body checks for the rest of the conjuncts' presence in the data base<sup>2</sup>. If all of them should happen to be present, the agent applies the closure computed above (to no arguments of course) and exits. If the other items are not present, the agent composes a second-order agent. This new agent is also an IF-ADDED whose invocation pattern is an instance of the second conjunct of the original disjunct. This instantiation is prompted by the bindings of various pattern match variables that prevail in the scope of the first order agent's body. The body of the new agent first checks for the remaining agent conjuncts (just as the first order agent did). If the check is successful, the last closure is applied. Failing the test for the rest of the conjuncts a third-order agent is produced using a new closure, and so on.

Much care must be taken in the writing of ENTAILments since it is possible to invoke the same code body (or at least equivalent code bodies) more than once. In any case, the agent AGENT-156 introduced in section 9.1 is a slight misrepresentation. The agent looks more like the value of

(LIST

<sup>1</sup> A similar closure must, in fact, be performed for compiling the determiner IF-NEEDEDs.

<sup>2</sup> The check of their presence must be done modulo the appropriate bindings of various match variables mentioned in the original ENTAILment pattern. The complication introduced by variables is easily taken care of, but not very interesting.

```

*IF-ADDED
(PAT-CLOSURE !" (CARRIER-COMPONENT ,~(SIGNAL ANT PF-3) ,ccl) fr)
;A check for other conjuncts would go here if there were any.
(CLOSURE ' (CLAMBOA () (COND ((EQUAL ... ) ... ))) fr)))

```

where PAT-CLOSURE is analogous to CLOSURE but closes a pattern with respect to the present bindings of pattern matching variables in contrast to  $\lambda$ -variables.

#### 10.4 Needs and provisions.

In radio plans, as in all plans, there are activities that are essentially ancillary to the "main line" of the plan. Sussman [1975] and Goldstein [1974], for example, distinguish "main steps" and "prerequisites." WATSON's plan definitions make analogous distinctions. The expansions of s-187, s-188, and s-189 yield the second kind of commentary. These expansions are respectively

```

(NEEDS (BIAS (ACROSS ~(PWR ★)))) ,
(NEEDS (BIAS (ACROSS ~(PWR ★)))) ,

```

and

```

(PROVIDES (BIAS (ACROSS ,~(PWR ★))) (FOR ,~(RF-SECT)))
(PROVIDES (BIAS (ACROSS ,~(PWR ★))) (FOR ,~(AF-SECT))).

```

The NEEDS expressions attest to RF-SECT's and AF-SECT's respective needs for a power bias, while the PROVIDES expressions indicate that PWR-SPPLY has taken the responsibility of fulfilling the mentioned NEEDS. These expressions have two kinds of compilation applied to them. The first kind of compilation produces data base items as commentary on the plan-fragment PF-3, presently being instantiated. This commentary is semantically equivalent to the NEEDS and PROVIDES expressions as they stand. The second kind of compilation is motivated by expectations to be met. That is, if a part in a plan has associated NEEDS (PROVIDES) commentary, then it is expected that some other part appear with complementary PROVIDES (NEEDS) commentary. This complementary match is monitored by IF-ADDEDs compiled for the express purpose of detecting the appearance in the data base of a PROVIDES (NEEDS) complement.

### 10.5 Further expansion of the RF portion of the GR-78 design.

Recall that the RF-SECT part of the top-level plan-fragment was claimed to be an instance of the plan SUPER-HET-RF<sup>1</sup>. The instantiation of the RF section plan shown in section 10.1 was actually incomplete. An accurate statement of the instantiation is

```
(INSTATIATE (RF-SECT PF-3) SUPER-HET-RF (CONTROL-BINDINGS (TUNING TUNING))).
```

The additional list APPENDED to the earlier expression is an identification of control variables in the RECEIVER plan with control variables in the SUPER-HET-RF plan. This modification to the instantiation will eventually result in the compilation of a (CTL-IO-WITH ... ) item to be ADDED to the data base. The definition of the SUPER-HET-RF plan is

```
(DEFPLAN SUPER-HET-RF CASCADE
  (PORTS (INPUT (ANT-IN INFO) (PWR POWER))
    (OUTPUT (DET-OUT INFO))
    (INTERNAL (RF-CNV INFO) (CNV-IF INFO) (IF-DET INFO) (AGC1-BIAS CTL)
      (AGC2-BIAS CTL)))
  (CONTROLS TUNING VOLUME RF-GAIN)
  (PARAMETERS INTERMEDIATE-FREQUENCY)
  (PARTS RF-AMP CONVERTER IF-STRIP AGC1 AGC2 DETECTOR)
  (GOALS
    ((DEMODULATE ... )
      (PURPOSES
        (CONVERTER (MIXDOWN ... ))
        (DETECTOR (DEMODULATE ... )))))
    ((SENSITIVITY ... )
      (PURPOSES
        (RF-AMP (AMPLIFY ... ))))
    ((SELECTIVITY ... )
      (PURPOSES
        (IF-STRIP (ADJACENT-CHANNEL-REJECTION ... )))))
  (RULE ... )
  (CASCADE
    (RF-AMP (ANT-IN (INPUT AGC1-BIAS) (OUTPUT RF-CNV))
      (NEEDS (BIAS (ACROSS ,~(PWR ★)))
        (NEEDS (BIAS (ACROSS ,~(AGC1-BIAS ★))))))
    (CONVERTER ... )
    (IF-STRIP (PWR (INPUT CNV-IF AGC2-BIAS) (OUTPUT IF-DET))
      (NEEDS (BIAS (ACROSS ,~(AGC2-BIAS ★)))) ... )
    (AGC1 ( ... )
```

<sup>1</sup> This section is mostly for the sake of continuity in filling out the design of the RF portions of the GR-78. In so doing, some other features of plan definition semantics are introduced. Though these features have been implicitly utilized in the explanations of the scenarios, the details are not crucial to the sections which follow.

```

    (PROVIDES (BIAS (ACROSS ,~(AGC1-BIAS ★)) (FOR RF-AMP))))
  (AGC2 ( ... )
    (PROVIDES (BIAS (ACROSS AGC2-BIAS) (FOR IF-STRIP))))
  (DETECTOR ... ))).

```

Since port names are local to a plan, in instantiating a plan-fragment from a plan there is a problem in identifying the ports of the super-plan-fragment with those of the sub-plan-fragment being instantiated. Rather than making explicit identifications (as is the case in binding control variables), ports are simply identified by positional correspondence between the ports of parts mentioned in the CASCADE sub-expression of the super-plan with the ports named in the PORT sub-expressions of the sub-plans with which those parts are identified. The SUPER-HET-RF plan is incomplete as it stands. The features that I want to illustrate do not depend on the burdensome syntax necessary to filling out the plan definition. The first thing to be noticed is the fact that one of the GOALS -- (DEMODULATE ... ) -- corresponds to PURPOSE noted for the RF-SECT part in the RECEIVER plan. Another feature to be noticed is that goals are decomposed into purposes. This decomposition explains what various plan parts do in order for the plan to successfully complete its intended task. Finally (if it is not already evident), cascade connectivity among the parts mentioned in the cascade is implicit in the fact that port names appear in more than one part's port specification. Incidentally, a canonical port name is generated whenever a port is not identified with a port in the super plan-fragment.

Descending the plan-fragment tree, we next look into the realization of RF-AMP in the last plan(-fragment). As usual

```

(INSTANTIATE
  (RF-AMP PF-19)
  MB-RFA
  (CONTROL-BINDINGS (TUNING BAND-PASS-FREQ)))

```

Instantiates

```

(DEFPLAN MB-RFA CASCADE
  (PORTS (INPUT (RF-IN INFO) (PWR POWER) (AGC CTL)) (OUTPUT (RF-OUT INFO))
    (INTERNAL (INFIL-AMP) (AMP-OUTFIL INFO)))
  (CONTROLS BAND-PASS-FREQ RF-GAIN)
  (PARAMETERS MAX-BAND-PASS)
  (PARTS INPUT-FILTER AMP OUTPUT-FILTER)

```

```

(GOALS
  ((AMPLIFY ... (VARIABLE-BY AGC) (CONTROLLED-BY RF-GAIN)) ... )
  ((TRANSMIT (BANDWIDTH MAX-BAND-PASS) (CENTERED-AT BAND-PASS-FREQ)
    (BETWEEN RF-IN RF-OUT))
    (PURPOSES
      P1.. (INPUT-FILTER (TRANSMIT ... ) ... )
      (OUTPUT-FILTER ... ))))
(RULE ... )
(CASCADE
  (INPUT-FILTER ( ... )
    (NEEDS N1.. (ALIGNED (WITH BAND-PASS-FREQ) STAGGERED)))
  (AMP ... )
  (OUTPUT-FILTER ( ... )
    (NEEDS (ALIGNED (WITH BAND-PASS-FREQ) STAGGERED))))
(SUPPORTS ((N1) (P1))))

```

Notice that the AMPLIFY goal has subexpressions indicating that the amplification is changeable in two ways: it may be varied by an externally supplied signal, or it may be controlled by a user-manipulatable control variable declared in the plan. The externally supplied signal is identified with the signal coming in from AGCI in the super-plan fragment PF-19. If we were to realize the AMP part as a plan-fragment we would eventually see that the control variable, RF-GAIN, would be identified with a variable resistance. Moving down to the CASCADE description, we see that the two filters are in need of alignment with a frequency standard provided by the BAND-PASS-FREQ control variable. These two NEEDS expressions are the source of the ALIGNED commentary used in section 6.3. Finally, notice that some of the PURPOSES and NEEDS are labelled (e.g. 'P1..' and 'N1..'). These labels are utilized in the (SUPPORTS ... ) sub-expression, indicating that the satisfactory realization of the purposes is directly dependent upon the meeting of the needs. The items compiled from this SUPPORT commentary enabled WATSON (in the alignment scenario) to determine that the filters were failing to support the image rejection goal of the RF amplifier. Also, the alignment prerequisite is recognized as being somewhat special. Instead of compiling an IF-ADDED to check for its satisfaction, an IF-NEEDED is compiled, which gets invoked by the procedure TEST when the alignment property is examined.

## 10.6 Compiling caveats.

In both the scenario of the misbehaving automatic gain control and the scenario of the misaligned front end, the phenomenon of the caveat played a key role in localizing the failures in those situations. To understand the machinery that creates the caveats we will need to look into the plan for the CONVERTER part of PF-19 which is partially given by

```
(DEFPLAN CONVERT CASCADE
  (PORTS (INPUT (RF-IN INFO) (PWR POWER)) (OUTPUT (IF-OUT INFO))
    (INTERNAL (MIX-IN CTL)))
  (PARTS MIXER OSCILLATOR) ... ).
```

After instantiating CONVERT as PF-74, its MIXER part is in turn realized as an instance of

```
(DEFPLAN MX CIRCUIT
  (NODES ... )
  (PARTS ... 2G-FET ... )
  (PORT-PAIRS ... )
  (REQUIREMENTS
    ((CARRIER-REJECTION
      (HIGH-BAND <<<.,~(FREQUENCY CARRIER SIGNAL OSC-IN ★)
        -.,~(FREQUENCY CARRIER SIGNAL RF-IN ★)> 2>
        + <.,~(FREQUENCY CARRIER SIGNAL RF-IN ★)>>))
      (AT RF-IN))
    ((CARRIER-REJECTION
      (LOW-BAND <.,~(FREQUENCY CARRIER SIGNAL RF-IN ★)
        - <2 * <.,~(FREQUENCY CARRIER SIGNAL OSC-IN ★)
        - .,~(FREQUENCY CARRIER SIGNAL RF-IN ★)>>>))
      (AT RF-IN)))) ... ).
```

The feature of note in this plan is the REQUIREMENTS declaration. Its syntax is like that of a GOALS declaration. It is a statement of the conditions that must be met in order for the plan to work. In English it says that carrier whose carrier frequencies are images of the RF signal to be mixed down had better be rejected at the RF input port of the MX plan. REQUIREMENTS are like NEEDS in that they lead to the raising of expectations<sup>1</sup>. Unlike the expectations raised on declaring a not yet realized plan part, the declaration of NEEDS or REQUIREMENTS need not be satisfied at design assembly time.

-----  
<sup>1</sup> This expectation arises in this plan because the designer knows that he is going to employ a four-terminal device (2G-FET in the PARTS declaration) in a non-linear fashion to achieve the mixing goal of his plan. This means that some care must be taken in using the plan.

The object of compilation of the requirements is executable code that will check that the requirement is being met. So one of the requirements above leads to

```
(IF-ADDED C-ALARMIST-196 (CARRIER-FREQUENCY cc cf)
  "AUX" (locals-frame apro-part)
  (FOR (THE-ONLY 1) (sig) !" (CARRIER-COMPONENT sig ,cc)
    (FOR ALL (pf port) !" (PF-OBS-SIGNAL pf port ,sig)
      (COND
        ((AND
          (EQ (CSETQ apro-port port) PORT-197)
          (M/L-EQUAL
            ..~(.cf)
            (MINUS
              ..~(FREQUENCY CARRIER SIGNAL RF-IN PF-75)
              (TIMES
                2.
                (MINUS
                  ..~(FREQUENCY CARRIER SIGNAL OSC-IN PF-75)
                  ..~(FREQUENCY CARRIER SIGNAL RF-IN PF-75))))))
          ( ... ) ;Code to assign LOCAL's frame to be the frame of the
                  ; nearest enclosing invocation of LOCAL.
          (ADD !" (CAVEAT C-ALARMIST-196 (CARRIER-REJECTION ... ) PF-75))
          (GO 'caveat-entry locals-frame)))))).
```

C-ALARMIST-196 says that if any carrier-frequency is ever ADDED to the data base, check if it is the carrier frequency for a signal at a certain (canonical) port. If so, see if the value of the frequency makes it an image of the frequency of the signal at the RF input port of the mixer plan-fragment. If that is the case, declare caveat trouble to the invocation of LOCAL in whose scope the present method is running. Note that a very interesting generalization has occurred. The caveat alarmist does not ask about images at the RF input port of the mixer plan-fragment, but rather at any port whose canonicalization is the same as the canonicalization of the RF input port of PF-75. Hence, the alarm will go off when examining the signal at the input port of the converter plan-fragment (one level up in the hierarchy) rather than having to wait until getting down to the mixer plan-fragment. Before going off to interrupt LOCAL, the caveat records sufficient history so that LOCAL can tell why the caveat feels offended. A general feature of the compilation of caveats is that they are made applicable at as high a level as possible in the plan-fragment. This possible to do simply by observing when a requirement is hanging on an external artifact such as a port or control.



Before leaving the topic of caveat compilation, let's consider the caveat that was triggered owing to the misbiasing of the IF strip. The plan from which PF-65 was instantiated<sup>1</sup> has a requirement for its operation given by

```
(REQUIREMENTS ((BIAS (ACROSS RF-IN) (> ., ~ (CLASS-A-BIAS BJT))))).
```

This leads to the compilation of the method

```
(IF-ADDED C-ALARMIST-198 (PORT-BIAS sig pb)
  "AUX" (locals-frame apro-part)
  (FOR (THE-ONLY 1) (pf port) !" (PF-OBS-SIGNAL pf port ,sig)
    (COND ((AND (EQ (CSETQ apro-port port) PORT-199)
      (M/L-EQUAL ., ~ (,pb) ., ~ (CLASS-A-BIAS PF-65)))
      ( . . . )
      (CSETQ caveat-message . . . )
      (GO 'caveat-entry . . . ))))).
```

C-ALARMIST-198 is quite analogous to C-ALARMIST-196. It first checks to see if the bias just asserted in the data base is for a particular port. If so, it then asks if the value of the bias is within the tolerances laid down by the REQUIREMENTS expression. If the bias is not like the class A bias that is dug out of the data base, the trouble is reported to the nearest activation of LOCAL.

#### 10.7 Plans with optional uses.

Recall that in the scenario concerning the failure of C422 a procedure -- OP-SPEC-MATCH -- was invoked in order to determine if there might be another use of the plan for the first IF amplifier, IF1, whose required operating conditions would match the observed state of affairs at the AGC bias port. In this section I shall explain the mechanisms facilitating this matching process. First, let me suggest that it is reasonable to carry out such a match since the AGC is a control signal. Thus we may think of the rules for a plan as taking on somewhat different characteristics, depending of the kind of signal at the AGC port. Indeed we have already encountered plans whose rules had outputs depending on control inputs. Here we shall

-----  
<sup>1</sup> We shall be looking into this plan in greater detail in the next section.

see the conditional behavior of the plan's rule as explicit commentary external to the code compiled from the plans's rule<sup>1</sup>. The plan of the first IF amplifier is

```
(DEFPLAN NB-RFA1 CIRCUIT
  (NODES GND IN OUT BAS COL EMM AGC QBIAS)
  (PARTS ... BJT ...)
  (PARAMETERS CLASS-A-BIAS CLASS-B-BIAS)
  (PORT-PAIRS (INPUT (RF-IN (IN GND) VOLTAGE) (AGC-BIAS (AGC GND) VOLTAGE)
                (PWR (QBIAS GND) VOLTAGE))
              (OUTPUT (RF-OUT (OUT GND) VOLTAGE)))
  :
  (REQUIREMENTS
    ((BIAS (ACROSS AGC-BIAS) (> CLASS-BIAS)))
    v((BIAS (ACROSS AGC-BIAS) (< CLASS-A-BIAS) (> CLASS-B-BIAS)))
    v((BIAS (ACROSS AGC-BIAS) (< CLASS-B-BIAS)))
  (RULES
    ( ... )
    v( ... )
    v( ... )) ... ).
```

It is instantiated by evaluating (recalling that the IF strip plan-fragment is PF-63)

```
(INstantiate
  (IF1 PF-63)
  NB-RFA1
  (REQUIRING ((BIAS (ACROSS RF-IN) (> CLASS-A-BIAS)))) ... )
```

which realizes the IF1 part of the IF strip plan as an instance of NB-RFA1. The variety of the latter plan used meets the first optional requirement, hence uses the first rule. Now though the "type" of the plan-fragment is declared to be "class A bias," the caveat alarms, agents and determiners are compiled for each of the other possible types. Only those methods associated with the first type are activated in the normal course of events by ADDing them to the data base.

-----  
<sup>1</sup> The computational analogy that comes to mind is the use of type declarations in an interpreted/compiled language environment. What should the interpreter do with declarations that are really meant as advice to the compiler? Consider the example of a program that adds the values of two identifiers. A programmer might declare both identifiers to be of type integer for the compiler's benefit so as to cause the use of the integer addition function. When executing the code interpretively it seems that the interpreter might run "carefully," taking note of declarations, and explicitly checking the types of the objects assigned to the identifiers to make sure they met specifications. In any event, the type declaration effects the compilation of code that need not check for the integerness of the operands of the addition. Thus a COND is factored out of the generalized addition function, yielding integer addition.

The others are remembered with their histories of construction. The history -- an association among agents, determiners, and the caveats compiled from the determiners -- is necessary to keep the various methods in "good company." WATSON must keep in mind that a certain agent applies only in the context of a particular requirement's being met.

When OP-SPEC-MATCH was invoked, it was passed the name of the offended caveat as a parameter. A bit of manipulation of the data base via evaluation of FINDs shows the trouble to be

```
(FALSE (> .~(BIAS OBS-SIGNAL AGC-BIAS PF-65) 1.0).
```

By means of the caveat compilation history, WATSON knows that this condition offends the IFI plan's first requirement. From that point it is a simple matter of examining the other sets of requirements to see if any of them is satisfied by the prevailing circumstances. One -- the singleton set of requirements consisting of

```
(< .~(BIAS OBS-SIGNAL AGC-BIAS PF-65) .6)
```

-- is satisfied, a determination which can be made by fetching the value of the ob denoted by (BIAS ... ). Having decided that there is a set of requirements which are satisfied, OP-SPEC-MATCH returns the names of the methods that describe the behavior of PF-65.

#### 10.8 A few words about components.

COMPONENT plans are essentially circuit plans with no internal structure. Their external nodes are their terminals. For example, the resistor, R49I (see figure 2.12) is described by

```
(DEFPLAN V-COUPLE-RES COMPONENT
  (NODES (EXTERNAL (#1 #2)))
  (GOALS ((VOLTAGE-COUPLES (BETWEEN #1 #2))))
  (RULE
    (ENTAIL !" (DC-NV , , ~(#1 ★) . desc)
      (COND ((EQ (CAR desc 'UNCHANGED))
        (ADD !" (DC-NV , , ~(#2 ★) UNCHANGED))) ... ))
    (BUG 'OPEN
      (ENTAIL !" (DC-BV , , ~(#1 ★) . desc))))
```

which compiles into code indicating that an assertion of unchanged voltage at node #1 leads to an assertion of unchanged voltage at node #2. Note that there is additional descriptive information regarding the open circuit bug. Its ENTAILment code body is empty. Thus when the resistor hypothesized to be "opened" it will not respond to voltage changes at its #1 node and the current through it is known to be 0 A.

#### 10.9 Miscellaneous theorems.

Many of the obs generated by the instantiation of plans have relational properties that are unknown at the time of instantiation. Thus, for example, the VALUE-RANGE property of the amplitude descriptor mentioned in section 4.1 is not known at the time of instantiation of the top-level plan-fragment. Similarly plan parameters typically do not have their values around when their corresponding obs are created. For example, the intermediate frequency parameter is not known at the time of the creation of the plan-fragment for the IF filter part of the IF strip. The reason it can be referenced is that there is a chain of identifications of plan parameters. In order to support the FINDing of such properties, there must be mechanisms for tracing such chains to values. The mechanism in question is a class of IF-NEEDEDs called identification theorems.

Such a theorem is employed in the running of the caveat alarmist considered in the previous section. In order to test its own applicability the alarmist does an implicit evaluation of

```
(FIND (THE-ONLY 1) (val) !" (VALUE ., ~ (INTERMEDIATE-FREQUENCY PF-65 val))).
```

In fact, the ob to which ., ~ (INTERMEDIATE-FREQUENCY ... ) denotes no VALUE property presently in the data base. The following identification theorem, however, is applicable:

```
(IF-NEEDED ID-TH-200 (VALUE param val)
  "AUX" (param-up param-down)
  (FOR (THE-ONLY 1) (pf) !" (PF-PARAM pf param-name ,param)
    (CSETQ param-up param param-down param)
  :LP (FOR (THE-ONLY 1) (p) !" (PARAM-ID-WITH ,param-down p)
    (COND ((PRESENT !" (VALUE ,p val))
      (NOTE !" (VALUE ,param ,val))
      (EXIT)))
```

```

      (T (CSETQ param-down p)))
(FOR (THE-ONLY 1) (p) !" (PARAM-ID-WITH p .param-up)
(COND ((PRESENT (VALUE .p val))
      (NOTE !" (VALUE .param val)))
      (T (CSETQ param-up p) (GO 'LP))))).

```

This code looks up (down) the tree of parameter identifications until it finds an identified parameter with a value property. It then makes up a value property item for the original parameter ob and NOTES it.

## 11 Miscellaneous Details

The aim of this chapter is to clear up some details that have been misrepresented for the sake of clarity. I should also like to tie up the remaining loose ends in the explanation of WATSON's performance.

### 11.1 Scenario completeness.

In order to understand how the incompleteness demonstration of section 4.3 is carried out, a fuller explanation of the items in WATSON's data base is necessary. All items have names. So items that I have previously described as looking like '(item-body)' in truth look like '(item-name (item-body))'. This additional feature entails no interesting changes to the machinery as I have already described thus far. These names are used as handles for making comments about the items they are attached to; in particular, items can mention other items. The RESULT structures introduced in section 4.3 are fictitious. The real truth is to be had by considering the item named ITEM-201, which is the result of applying RULE-202 to the items ITEM-203 and ITEM-204. This relationship among the various items is indicated by the comment item

(C-ITEM-205 (RESULT ITEM-201 RULE-202 ITEM-204 ITEM-203)).

Of course the commenting item itself has a name, C-ITEM-205. Another kind of commentary that may apply to an item is an indicator of time of validity:

(C-ITEM-206 (RESULT-TIME item-name time-sequence-number)).

If an item does not have such a comment about itself, it is presumed to be applicable now.

Otherwise the item is interpreted as being applicable at some point in a sequence of times. We shall see this interpretation made use of shortly.

Another mis-representation I made is to be found in the PF-GOAL item of section 4.3. To understand this better we must look into the plan that was the source of PF-32, the oscillator plan-fragment. The relevant details of the plan are given by

```
(DEFPLAN OSC CASCADE
  (PORTS (OUTPUT (OSC-OUT INFO))
          (INTERNAL (OSC-IN INFO) (OSC-GAIN CTL) (AMP-OUT INFO)))
  (PARTS AMP TANK AMPL-STABL)
  (CONTROLS TUNING-FREQUENCY)
  (PARAMETERS IDEAL-AMPL AMP-GAIN)
  (GOALS
    ((STABILIZE ,~(AMPLITUDE CARRIER SIGNAL OSC-OUT ★)
      (AT ~(IDEAL-AMPL ★)
        (AGAINST (DECREASE ,~(AMPLITUDE CARRIER SIGNAL OSC-OUT ★))))
      (SCENARIO ... )
      (PURPOSES
        (TANK (TRANSMISSION (BETWEEN AMP-OUT OSC-OUT)))
        (AMPL-STABL
          (INCREASE AMP-GAIN
            (WHEN (DECREASE ,~(AMPLITUDE CARRIER SIGNAL OSC-OUT ★))))
          (AMP (AMPLIFY
            (BETWEEN OSC-IN AMP-OUT)
            (SET-BY AMP-GAIN)))))) ... ) ... )
```

which says that one of the goals of the oscillator plan is to stabilize the output amplitude (at some ideal amplitude) against a possible amplitude decrease. The PURPOSE commentary goes on to recount how the plan parts would behave if the amplitude at the oscillator's output were to decrease. The manufacture of a plan-fragment from the OSC plan leads to the compilation of the following items (among others):

```
(GOAL-207
  (PF-GOAL PF-32 RULE-37
    (STABILIZE ,~(AMPLITUDE CARRIER SIGNAL ... ) ... )
    (,~(TANK PF-32) PURP-208
     ,~(AMPL-STABL PF-32) PURP-209
     ,~(AMP PF-32) PURP-210)))
```

and

```
(PURP-208
  (PF-PURPOSE ,~(TANK PF-32) PF-32 GOAL-207 RULE-211
```

(TRANSMISSION ... )))

which show the intimate relationship between statements of goals and purposes of parts used to fulfill those goals. The specification of the scenario -- (SCENARIO ... ) -- is not shown because of the painful syntax necessitated by descriptions of time and causality. However, the scenario compiles into a large collection of items, starting out with

```
(RES-40 (< ~(AMPLITUDE SIGNAL AMP-OUT PF-32)
          ~(IDEAL-AMPL PF-32)))
(ITEM-212 (RESULT-TIME RES-40 0.))
(ITEM-213 (VALUE ~(AMPLITUDE SIGNAL AMP-OUT PF-32) ?))
(ITEM-214 (RESULT-TIME ITEM-213 0.))
(ITEM-215 (RESULT RES-40 INIT)).
```

These items state in brief that WATSON should imagine a situation in which the output amplitude is less than some optimal value as initialized at time 0. Observe that the RES-40 here is the true form of the item that made an appearance in section 4.3. The rest of the scenario is compiled into similar items stating the tie course of events in the oscillator's self-stabilization.

With this scenario in hand WATSON can investigate whether or not the scenario can be completed under the circumstances discovered during localization. Recall that WATSON wanted to know what would happen to the oscillator if the amplifier were not behaving according to its rule. The way he finds out is by making a fresh context (with all the oscillator parts' rules active except for that of the amplifier) and asserting all of the RESULT items that follow from the initialization rule INIT. When the firing of various rules ceases, WATSON takes each of the items in the scenario and looks for analogous items in the fresh context. If no such item is forthcoming from the latter, the scenario is said to be incomplete with respect to that item. In the case of Q301 there will be items indicating an increase in oscillator amplifier gain when the amplitude of the oscillator output decreases. This last shows the scenario to be incomplete, more particularly demonstrating that the amplifier will be downwardly unstable -- exactly as observed.

## 11.2 Consistency monitoring.



Recall that in section 4.4 LOCAL attempts to verify the consistency of the conditions at the terminals of the transistor, Q301. In light of the discussion above, the RESULTS mentioned in the course of carrying out the analysis are really named items of the form

```
(RES-56 (DC-BV ... ))
(RES-58 (DC-BC ... ))
```

with suitable commenting items about how they came to be. Each time one of these RESULTS is produced by an agent of a rule, the agent also ADDs to the data base an IF-ADDED to detect if any contrary items are ever asserted. RESULTS RES-56 and RES-58 were determined to be inconsistent by

```
(IF-ADDED CONSIST-216 (DC-BC ( ... ) spec )
  (COND ((NUMBERP spec)
    (COND (((NOT (M/L-EQUAL spec oldspec))
      (GO 'inconsistent localfr)))) ... )).
```

This method is made up as a closure by the same rule that generated RES-56. It says that if the current in the source/drain branch of Q301 is specified as a number, it had better be the same number as seen before. If it is not the same, the nearest enclosing activation of LOCAL is entered at an error tag. The remaining clauses of the outer COND test for other varieties of specification in consistency.

### 11.3 LOOPS.

Presumably, it is now clear how the stabilization scenario analysis carried out in section 4.3 is effected. There remains the question of how LOOPS knows to pick on the particular goal that carries the scenario. This is easy enough. Recall that LOOPS was invoked because it went around a loop in a cascade plan, discovering that every part had outputs consistent with its inputs. Referring to the PF-GOAL item in section 11.2 (GOAL-207 ... ) we note that the goal encompasses a number of parts -- namely the parts for which there was PURPOSE commentary

under the corresponding GOAL expression in the OSC plan. The heuristic used by LOCAL is to choose a PF-GOAL item that mentions all the parts that were passed through in going around the loop. LOOPS then uses the powered part heuristic to suggest a part failure, followed by a scenario completeness analysis to see what happens. As it happens there are two applicable goals here -- one describing upward stability, and one describing downward stability for the oscillator. The former must be rejected since it claims that the end result (when the oscillator amplifier is not working) is a larger than normal signal amplitude -- obviously false. Hence the latter goal and scenario are pursued.

A more complicated loose end to tie up is to be found in the techniques for loop breaking. None of the scenarios presented made use of loop breaking in the course of localization. It seems worthwhile, nonetheless, to present the essential details of how it would be done. For concreteness I shall use the feedback loop of which AGC2 is a part in figure 2.5. The shorting of C422 was found by WATSON using strict causal analysis. What if there had been a bug such that the IF strip, detector and AGC2 all had correct outputs given their inputs? Failing in achieving a quick solution via the powered part approach, WATSON decides to break the loop. The obvious question is "Where?" A heuristic answer is provided by the following observations. Feedback loops tend to be for the purpose of stabilizing some signal property along the forward path of the loop. Consequently, the point at which the feedback path enters the feed-forward path will typically be a control port. Control signals generally have simpler descriptions than do information signals at the same level in the plan fragment hierarchy. Hence it would seem that a reasonable place to break the loop is at the re-entry point.

Having chosen a *break-point*, WATSON must decide how to do the break<sup>1</sup>. WATSON

-----  
<sup>1</sup> The programming metaphor is intentional and appropriate. The problems of debugging electronic circuitry -- especially RF circuitry -- are not unlike those encountered in the debugging of real-time programs. Monitoring the behavior of such programs (break-points being a kind of monitor), unless done with care, can change the behavior. Thus software break-points that replace the programmer's code with some standard "parenthetic" code may cause timing bugs to appear or disappear. In such circumstances the programmer may have to carefully tailor some parenthetic code that minimally disturbs the normal course of events so as to be able to locate his bug. Of great aid in such circumstances are hardware "instruments" like the MIT-AI PDP-10's

distinguishes two kinds of loop breaking -- *open* and *closed*. The closed break *injects* a signal at the breakpoint without modifying the circuit's physical connectivity. (The purpose of the loop-breaking is to distinguish whether the problem is in the forward or reverse branch of the loop.) It turns out that the control port between the IF strip and AGC2 is a voltage port. Consequently it should be straightforward to manipulate the signal there without a physical break. Before setting about supplying such a signal there WATSON is careful to estimate the output impedance of the signal source at the control port<sup>1</sup>. This value is compared with output impedance of the instrument used to supply the signal. If the output impedance of the normal source is not substantially greater than that of the independent source, a physical break will be called for despite the fact that the port of interest is a voltage port. Because of the measurements problems for current ports, supplying an externally controlled signal at such a port always necessitates a break. A physical break requires an estimate of the input impedance of the plan-fragment at the output side of the break-point. An equivalent impedance is used to tie off the input side of the port after breaking.

Having decided what kind of break to do, WATSON must specify a signal to be supplied. This is fairly straightforward. The way the various parts on the loop were determined to have correct outputs given their inputs was by pushing the observed inputs of each the parts through their respective rules. Consequently the expected input at the AGC2/IF strip interface is a description of the signal WATSON wants to supply independently. This description may not be quite detailed enough. For example, WATSON may be localizing at a sufficiently high level in the plan-fragment hierarchy such that details like expected signal amplitudes may not be in evidence. But these can be estimated by descending into the plan-fragment structure and fetching the

-----  
MAR device, with which it is possible to monitor certain kinds of program events without disturbing the program.

<sup>1</sup> Such an estimate may be done by reading the value of the output impedance of the plan-fragment supplying the signal, or computing it by applying simple analytic techniques to that plan-fragment, or by asking the human assistant.

maximum and minimum allowable signal amplitudes and computing a medium value just as was done in setting up test situations.

#### 11.4 Plan flattening.

The final detail concerns the alignment process described in section 6.3. Namely, what is the nature of the flattened plan produced and manipulated by ALIGN? The flattening process is, in fact, the same as the one employed by LOCAL in rationalizing the output of the IF strip plan-fragment. Recall that flattening begins at some root plan-fragment (the top-level plan-fragment in the alignment case, and the IF strip plan-fragment in the case of C422) and raises some distinguished collection of sub-plan-fragments to the top level. Section 6.3 described the essentials of this recursive process. (The alignment case has a different terminating condition from the case of C422.) What is not so clear is the following: After the plan is flattened many of the items in the data base are no longer applicable. This is because they are assertions about measurements which are no longer valid (as in the alignment case) or they are assertions about expected signal descriptors which are no longer valid because the rules giving rise to those expectations are no longer in operation. Both classes of assertion in fact only exist in the context of the invocation of LOCAL in which they were measured or deduced.

The solution then seems clear. First sprout a new context in parallel to the one in which the root plan-fragment became active. Next activate in the new context the rules for the plan-fragments that have been raised. Remembering that the signal structures (both expected and observed) are common to all the plan-fragments meeting (as parts in an immediately superior plan-fragment) at a port, the signal structures for the plan-fragments external to the raised plan-fragments are then patched in as signal structures for the plan-fragments meeting those external plan-fragments. This takes care of invalidating old information.

In the case of plan-flattening for the purposes of rationalization, it is safe to bring

along the old measurement results to the flattened plan context. In the case of the plan-flattening for alignment, the old measurement results are invalid. In the course of alignment several mutually incompatible sets of measurements may be taken. This problem may be resolved by making RESULT-TIME commentary on each of the measurements items. Each time an alignment adjustment is made the "clock" is updated to show a new time. All measurements in the flattened context showing the old time will then be invalid.

## 12 Conclusions

WATSON has been shown to be capable of solving a particular collection of non-trivial problems in the realm of electronic trouble-shooting. There is a temptation to conclude this report with arguments for the efficacy and generality of WATSON's methods for his domain and leave it at that. Instead, I shall encourage the drawing of a more profound moral: that understanding and reasoning about complex machines is facilitated -- indeed, may only be possible -- by the guidance provided by hierarchical summaries of lower-level details.

### 12.1 The underlying concepts.

WATSON's basic strategy is that a part's broken-ness in most cases should be determinable solely from the local evidence supplied by its input/output behavior. Let's assume for a moment that the assumption of this strategy were universally true -- which is not the case -- and review how this strategy (used in an unstructured fashion) would lead to a quagmire of computational details. Using the bare circuit diagram and basic voltage/current models of the components contained thereon, predicting the behavior of a radio receiver under the influence of an electromagnetic field impinging upon the antenna requires computing the propagated effect on every resistor, capacitor, inductor, transistor, etc. in the receiver. This "flat" view of a radio receiver of moderate complexity incurs an enormous number of computations whether we choose to model the receiver by classical differential equations (a declarative model), or by the procedural (imperative) models (of components) that WATSON actually employs.

The "fix" to this unpleasant situation is to be found by recalling how the receiver

design came to be in the first place. The top-level goal was not to engender the appearance of a particular time-varying voltage across a resistor R1 in the face of a time-varying voltage across another resistor R2. The designer had some considerably more abstract goal in mind, as must be the case if there is to be any general notion of a radio receiver. Retracing the designer's thinking<sup>1</sup>, we see that the top level goal was to recover the information from an abstract coding scheme called modulation. Thus the main step of any radio plan must be detection. Unfortunately, realizable detection schemes are of necessity inefficient. Thus a weak signal might not be directly demodulatable to a measurable signal. Making the received signal bigger before applying detection should fix that problem. Still, the signal gotten out of the detector might not be able to drive an audio transducer. This difficulty can be overcome by audio amplification. In this manner there is quickly built up a hierarchy of modules, some of which are solutions to problems posed at higher levels, some of which are prerequisites for solutions, and so on.

The design process imposes an abstract hierarchical structure on the concrete radio. Each node in the hierarchy carries descriptive commentary in a language and at a level of detail appropriate to its rank in the hierarchy. Thus at one level we might see an account of detection consisting of transforming one variety of abstract signal description into another, whereas at another level of detail, we see a voltage rectifying network with a certain DC bias, followed by a low-pass filtering network. Using this hierarchical description, WATSON can achieve a kind of logarithmic search of circuit components in the task of isolating the broken component.

The fact that the computational complexity of predictive reasoning is considerably reduced by the imposition of an abstract structure is by itself significant. As we have already seen, this hierarchical framework provides another service at least as important as that of inferential simplification. As I mentioned earlier, WATSON would like to believe that a part's broken-ness can be determined completely from local phenomena: look at its inputs, predict the

-----  
<sup>1</sup> Of course, this retrace concerns not only the thinking of a single designer, but the "historical" designer as well.

outputs that should ensue, and check the predicted behavior against the actual behavior. The ability to do this is predicated upon the ability to achieve a goal via some *causal* plan. Causality says that the behavior of a part is determined solely by the state of the world as determined by the parts which in some abstract sense operated "before" our distinguished part. A system exhibiting such a causal structure is amenable to a failure localization technique -- back-tracing -- that is a direct consequence of the above notion of 'before'. A fly in the ointment is, of course, that electronic circuits incorporate at least one "control structure" -- feedback -- and other features as well that undo simple causality. Feed-back engenders situations like the one encountered in the case of Q301, in which a loop is circuited wherein every part's output seems reasonable in terms of its inputs, but the oscillator is in fact not functioning. As it turned out, localizing the failure required a more global view of the interactions of the oscillator's parts. But such global analyses seem antithetical to the desire to carry out strictly local reasoning. Or are they?

The trouble is that global analyses may sometimes facilitate the solution of a tough problem. Indeed, the solution may only be possible through global analysis. How is the desire for a problem-solving strategy based on (local) causality to be made commensurate with the occasionally non-local mechanisms. The answer is to be found in the observation that one plan's 'global' is another plan's 'local'. In other words, global problem-solving strategies should be applied using local direction. The invocations of LOOPS and ALIGN are obvious instances of essentially global reasoning being triggered by local phenomena. Localizations are achieved by appealing to knowledge global to the parts localized to. The hints suggesting the use of a global methods come from knowledge attached to the plan containing the parts.

Interestingly enough, the hierarchical plan structure suggests (admits) not only "upward" appeal when a further level of localization seems otherwise impossible, but "downward" appeals as well. Although that the initial motivation for a hierarchy is to keep unnecessary details out of sight (hence out of mind), sometimes those details are necessary, as when choosing an appropriate input test signal for the receiver. The only way to determine an acceptable amplitude for such a



signal is to look at the specifications of the RF amplifier sub-circuit. A more interesting version, still, of such downward appeals is to be found in the caveat construct. The caveat encountered in the case of C422 was used to provoke a downward search for details. At first glance it would seem that caveats should be entirely local to the plans that engendered them. But by allowing the caveat in the case of C422 to be visible at the IF strip level as well as at the level of IF1, the IF strip's unexpected output became justifiable. The caveat brought more detailed knowledge to the surface at just the right time. In compiling the caveat, the determination of the level at which the caveat was to become visible was based on a local structural phenomenon. The caveat comes through at the first level at which one plan's external port becomes another plan's internal port.

## 12.2 The machinery embodying the concepts.

WATSON is a rather complex program composed of several distinct mechanisms. Some of the more important mechanisms were surveyed in the overview given in chapter 3. Here we review those mechanisms with the hindsight provided by the detailed discussion of the scenarios.

WATSON's principal agent of control is the procedure I have referred to as LOCAL. LOCAL is typically entered from the top in the style of the standard function call, having been passed a number of parameters, the most important of which are the (local) plan-fragment suspected of containing the locus of failure, and the reason for that suspicion. A typical reason for the indictment of a plan-fragment will be incorrect output(s) given its input(s). LOCAL's assigned task is to further narrow the focus of attention to some part contained in the local plan-fragment.

To carry out this task a number of methods are at WATSON's disposal. The first question asked is whether or not any abstracted bug is already known whose recorded symptomatology matches the observed input/output behavior. Such bugs are to be found attached to the plan from which the plan-fragment currently of interest was instantiated. If a bug with the right symptomatology is found, its abstraction will indicate a sign to be tested for. The sign stands in

relation to one of the parts as the symptom stands with the containing plan-fragment. Of course, it may be that there is no abstracted bug applicable to the prevailing symptomatology.

In that case other localization methods are brought to bear. The most important of these is back-tracing. As mentioned in the last section, this method relies upon the essentially causal nature of many of the plans used in radio designs. Since a plan's parts can be viewed as depending only upon their immediate (in time and space) inputs, the obvious procedure of moving along the signal path, searching for the first part whose outputs are not compatible with its inputs, should uncover the source of the problem.

Back-tracing too may fail to get the job done. This may be due to global control structure in the plan -- like a feed-back loop -- or an insufficiently well-defined notion of causality -- the synergy of cascaded filters or the voltage/current models of some of the CIRCUIT plans we have seen. Some of these were taken care of by experts like LOOPS and ALIGN. Another solution method was to consider *a priori* probabilities of failure coupled with the prediction of the AC and DC consequences of such failures.

The principal data to which the problem-solving control structures above are applied are plan-fragments. These are best understood as compiled (particular) instances of (general) plans. Each plan-fragment in a design has a corresponding plan. Each part of the plan has a corresponding realization as a plan-fragment in the design. Thus the skeleton of a design is a hierarchy of plan-fragments, partially ordered by a "part of" proposition. Aside from parts, plans have a number of distinguished features, principal among which are *controls*, *ports*, *nodes*, *parameters*, *circuits*, and *cascades*. These indicate various kinds of external and internal interfaces among parts and how the parts are topologically organized with respect to those interfaces. A design may be further distinguished by the mutual identification of features at various levels in the plan-fragment hierarchy.

In addition to the "connective tissue" above, plan-fragments also bear causal and teleological commentary indicating the nature and purpose of their workings. This commentary

includes rules which model their signal or voltage/current processing behavior. A plan-fragment may have a goal indicating what function it serves. Similarly, a plan-fragment may have a purpose within some containing plan-fragment. A purpose -- which must match some goal -- indicates the function being served within some larger context. Finally there are a number of kinds of commentary specifying prerequisites that must be satisfied in order for a particular plan-fragment to be operational. Included in this class of commentary we find requirements, needs, and provisions.

Successful representation of causal mechanisms requires not only descriptions of behavior of individual parts, but descriptions of the parts interactions as well. The formal structures that serve as media for such interactions are signals, branch (node) voltage descriptions, and branch current descriptions. Changes in the descriptors making up these data structures may activate various propagation rules. Conversely, rules may induce changes to the descriptors. Causality is captured by this interplay of rules and descriptors (with suitable interpretations of time-order and event-order).

In an informal discussion of WATSON's performance it is convenient to think of plans and plan-fragments quite interchangeably. After all, the latter are simply instances of the former. There are many potential designs. Designs admit the possibility of both multiple instantiations of a given plan (as different plan-fragments), and the realization of different parts as the same plan-fragment. These aspects of designs (among others) require a richness of descriptive power in the design assembly specification. A design (indeed all of the structures manipulated by WATSON) may be thought of as a collection of named assertions. These assertions attest to the truth of some proposition concerning formal objects called obs. The assembly process commences by mentioning a plan for the top-level plan-fragment of the receiver design. The assembler reads the plan and compiles appropriate assertions that are the realizations of ports, controls, parts, topology, etc. that the plan specifies. Assembly engenders the compilation of methods that guarantee the meeting of certain criteria for the plan-fragment's successful operation as

formulated by the originating plan. There are also compiled agents and determiners that will be the procedural embodiment of function and causality for the newly minted plan-fragment. Having made a top-level plan-fragment, the assembly process may be continued by identifying a certain plan with one of the parts of the plan just instantiated as a top-level plan-fragment. This identification causes the instantiation of yet another plan-fragment together with its associated assertions and CONNIVER methods. A linking assertion establishes this latter plan-fragment as the embodiment of a part in the top-level plan-fragment. This process of expansion of plan parts as instantiations of other plans may be continued indefinitely. While carrying out this expansion, there is also the possibility of doing various "inter-level" identifications of plan-fragment attributes.

The assembly process yields a structure that is the focus of WATSON's causal and qualitative reasoning. The capability for causal reasoning stems from the ability to answer questions about the output behavior of the radio given certain inputs and the specifications of its control settings. The essence of qualitative reasoning is WATSON's ability to examine the radio's behavior at many levels of detail. At one level of detail, for example, WATSON sees that an RF amplifier's output amplitude is much larger than its input amplitude. Whereas from another point of view he might see that the incremental voltage gain across the amplifier is precisely 21.5 whenever the input voltage is between 1.0 and 1.2 V. Causality is represented implicitly by the agents that propagate signals through the receiver design, and explicitly by determiners that indicate the causal antecedents of particular signal descriptors.

Supplementing the receiver-specific knowledge contained in a design is a wealth of general knowledge incorporated in the various theorems realized as IF-ADDED or IF-NEEDED methods. These include theorems concerning the accessing of values of parameter and control obs through upward and downward identification, the accessing of signal descriptor values through upward identifications and "across-port" identifications, and theorems concerned with generating "typical" values for signal descriptors of the variety utilized in validating a

COMPLAINT. This last group of theorems relies heavily on VALUE-RANGE commentary explicitly attached to plan-fragments, or alternatively on local circuit-level deductions of the kind reported by Sussman, Stallman and Roylance.

### 12.3 Some universals of problem-solving in deliberate systems.

At the outset of this chapter I suggested that the lessons to be learned from WATSON reached beyond the domain of the trouble-shooting of radios. My thesis is that all deliberate systems -- systems designed to achieve some end -- are best explained by a hierarchical decomposition into sub-modules. As I have previously observed, such a hierarchy is historically representative of the design process<sup>1</sup>, causality in the machine can be understood in terms of the local interactions of the parts at various levels of detail in the hierarchy, and, finally, various bodies of expert knowledge can be brought to bear based on features observed locally while traversing the hierarchy. To make the universality of these methods clear I should like to return to a class of machines that I alluded to in the introductory chapter. I shall consider a problem in the trouble-shooting of an automobile.

Let me preface the scenario that follows with a disclaimer. I do not intend to suggest that WATSON's methodology is trivially transferrable to the micro-world of auto mechanics. On the contrary, there are many interactions in an automobile design<sup>2</sup>, the representation of which is not at all obvious at my present stage of thinking about the problem. For example the top-level plan(-fragment) of an automobile has three parts, or major subsystems: carriage, power-train, and steering. Obviously these systems overlap in their containment of certain components -- like tires. The difficulty is that I do not know how to specify the interactions between the carriage and the power train, for their interactions are many and qualitatively distinct. Hence I shall focus my

-----  
<sup>1</sup> That a complex machine could have been designed at all hinges on this hierarchical understanding of what is to be done.

<sup>2</sup> 'Design' is used here in the same technical sense as in the WATSON program.

attention on the engine part of the power train plan, so as to best elucidate the analogy with the representation and repair of radios.

Suppose the complaint about the automobile is that it stalls inexplicably after a few minutes of faultless running<sup>1</sup>. Since we know that the engine ceases to run when disengaged from the rest of the power train we might immediately localize to the engine. Nonetheless, it is instructive to carry out a top-down analysis, so let's look at the back-trace procedure that would have gotten us to the engine anyway. The top-level plan-fragment (see figure 12.1) has two inputs -- air and fuel, the latter being buffered in some fashion -- and an output -- a torque delivered at the driving wheels. The output torque can be converted into forward motion of the vehicle if certain constraints are met. The nature of the complaint is such that this torque is delivered for a while and then ceases. So we check the inputs: the fuel tank is full and the air supply meets the usual atmospheric specifications at sea level. Notice that in contrast to radios, the inputs and outputs in the case of an automobile may take on many modalities -- fluid flow, torque (as we have just seen), heat flux, etc.<sup>2</sup> Another complicating feature is that "memory," as embodied in the fuel tank for example, is quite common in mechanical devices. This feature is also common in electrical circuits: electromagnetic energy stored in a parallel tuned circuit, the heat energy that can accumulate in a radio (potentially resulting in the receiver's failure), and the potential energy stored in a battery. The fact is that WATSON has gotten away with successfully explaining radios (for his purposes) without resorting to the "storage" metaphor that constitutes an integral part of a deep understanding of electronics. In any event, an understanding of accumulation and capacity seems unavoidable for successful comprehension of mechanical devices.

The back-trace proceeds as follows: keeping our finger on the output port provided by

-----  
<sup>1</sup> Note that the formulation would require a time course description. The representation used for describing stabilization in the oscillator would do.

<sup>2</sup> Actually, we might think of abstract signals, branch voltages, branch currents, node voltages, and loop currents as different input/output modalities. Also control settings constitute a kind of mechanical input.

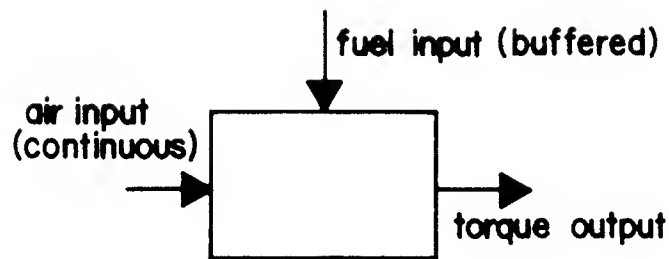


Figure 12.1 -- The top-level plan of an automobile.

the driving wheels we expand the top-level plan-fragment into its subsidiary parts and note that it is the part called the power train (see figure 12.2) that outputs to the port indicated by our finger. Note that the fuel actually gets buffered in the carriage part. This is a result of the fuel tank's being an integral structure (for the purposes of rigidity) with the carriage. Consequently, from the power train's point of view, fuel flows in continuous supply from the carriage. In the usual fashion we verify that the inputs to the power train are reasonable, but that the principal output -- torque at the wheels, -- is faulty<sup>1</sup>. We expand the power train part into the plan of figure 12.3 to further the localization of the fault.

Before continuing the localization, however, I should like to point out another complicating factor in automobile plans, the phenomenon of coupling. In radio plans, WATSON represented ports as passive constructs that were simply anchor points for the recording and measuring of certain kinds of information. Indeed one might argue that an AC coupling capacitor should be viewed as part of the port between two stages, but it seemed adequate to assign the capacitor to a plan-fragment of its own whose input and output ports coupled the two stages in question. Coupling, like storage above, plays an important role in the deep understanding of deliberate artifacts. Again WATSON has skirted the issue (successfully) by

-----  
<sup>1</sup> I should mention that I am again assuming the availability of all instrumentation that is feasible, but perhaps not available to the average auto mechanic. In particular, for the present scenario, I am assuming that the vehicle is attached to a road dynamometer for the purpose of making measurements at the final drive.

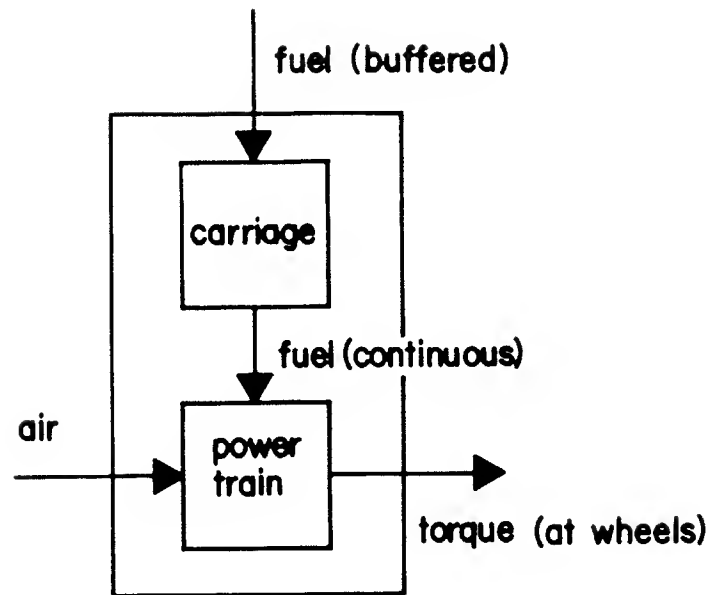


Figure 12.2 -- Expanded top-level plan.

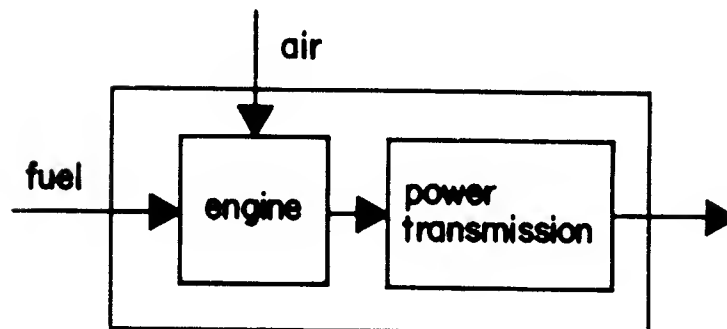


Figure 12.3 -- An expanded power train

various forms of subterfuge. An understanding of coupling is at the root of the electrical engineer's interest in impedance matching. Indeed, tuned couplings,  $\pi$ -networks, and transformers are coupling mechanisms used for achieving a variety of matches. Mechanical coupling can be done through many kinds of ports. Consider the constant velocity universal joint that typically interfaces the drive shaft to the gear box. If we assign the universal joint to its own plan-



fragment, we observe the fact that an integral part of the joint happens to be the U-shaped end of the shaft. I think that it might be more appropriate to represent coupling in the port itself with commentary that that port couples rotation in one plane into rotation in another plane. We might even coin a new name for such an "active" port. Call it an *interface*. In fact, we can readily suggest many such interfaces: reduction gears, bevel gears, fluid-coupled turbines, friction clutches, etc. All of the latter are idality preserving. There are transducing interfaces too -- like crankshafts -- that change linear motion into rotational motion (or force into torque).

To continue with the localization, the engine is expanded into the plan of figure 12.4. Back-tracing from the torque output, we find inputs from the cooling, electrical, and carburetion subsystems. Notice that the input ports bringing water and oil from the cooling subsystem are associated with water and oil output ports in a closed loop, indicating that the net flux of these materials into the motor is zero<sup>1</sup>. In any event, these flows are determined to be reasonable. The ignition (and carburetion signals too) are time multiplexed, with time slices being distributed among each of the cylinders. Thus at this level of detail in the plan-fragment hierarchy, it need not be known how many cylinders there are. Checking the input from the carburetion system, a caveat alarm is triggered.

The alarm has to do with the fact that the engine runs in three distinct temperature ranges. In the normal operating temperature range, the fuel/air mix requires more air (volumetrically) than does the mix when the engine is cold. There is presently too little air relatively, and forward reasoning shows that the motor should stall, causing the output torque to become zero as observed. The next order of business is to note that the responsibility for the correct fuel/air mixture lies with the carburetion subsystem. Looking into the carburetion subsystem reveals that the temperature/voltage transduction interface between the motor and the

-----  
<sup>1</sup> This is a kind of "alignment" property. In fact in certain racing engines it is literally that since the volumetric flows in the two arms of the loop are separately controlled, but must track one another. This follows from the apparently different flux rates due to frothing (which the designer typically wants to eliminate) and heat expansion.

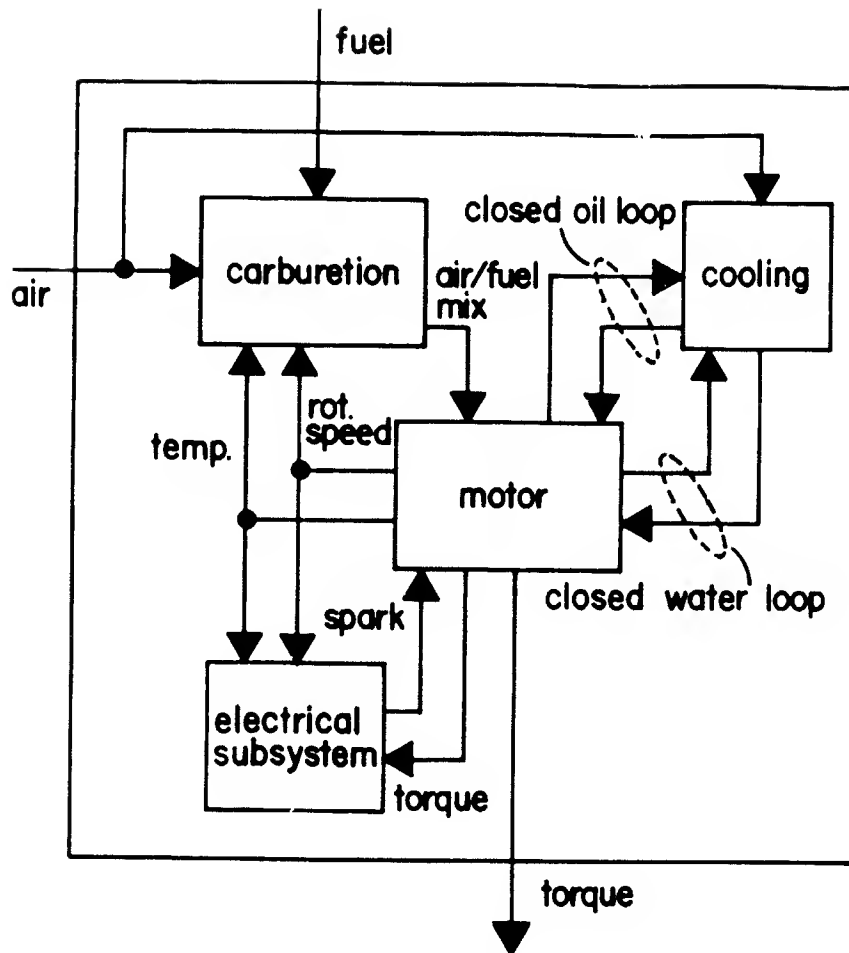


Figure 12.4 -- An expanded engine.

electronic fuel injection embedded in the carburetion subsystem shows that the voltage developed by the transducer does not correspond to the motor temperature, hence the faulty air/fuel mix. This completely localizes the problem.

The analogy with representation and reasoning in the micro-world of radio receivers should be self-evident. Of course I have conveniently glossed over many technical problems, like the details of interfaces, the loop breaking techniques peculiar to the plans of this micro-world, the descriptions of various signal modalities, signal measurement, etc. Still, the structure of

mechanism representation and failure localization is fundamentally the same. The reader may balk at the notion that a real mechanic actually does such a recursive localization. Indeed for designs that the mechanic is familiar with, he may immediately jump to a conclusion -- like looking at the fuel flow to the motor. The ability to make such leaps is a property of having achieved considerably richer abstractions of bugs than those proffered by WATSON -- but that is another thesis. I should reply that although the recursive localization is not applied top-down in familiar designs, it is so applied in unfamiliar ones. Just take your four-rotor Wankel-powered car to your friendly local service station sometime and see!

#### 12.4 WATSON's shortcomings.

The major thrust of my research in designing WATSON has been directed at the creation of control structures adequate to providing the hierarchical control of local reasoning that underlies his understanding of complex receivers. The emphasis on control structure has unfortunately left the representational mechanisms wanting in several respects. One major complaint that I have is that his plans are not sufficiently "elastic." The kind of elasticity I have in mind is suggested by Minsky's frame paradigm [Minsky, 1975]. His frames have slots with fillers that may be tightly or loosely bound to whatever degree seems appropriate. What do I mean by all of this? Recall that in chapter 10 I introduced a plan called SUPER-HET-RF. It demanded the existence of a number of parts: an RF amplifier, a converter, an IF strip, a detector, and a pair of AGC's. In fact, the "essence" of a superheterodyne RF section does not include automatic gain control. A much better plan representation would have demanded the first four parts, and suggested that one or both AGC's might be present. Thus the plan is "stretched" to cover a number of possible realizations of an RF section (all extrinsically the same).

Another weakness with WATSON's plans *vis à vis* representing more powerful "radio

frames" is that plans presently provide little aid in solving the schematic parsing problem<sup>1</sup>. The only way a plan may be seen in terms of more detailed plans is through the vertical expansion of plan-fragments in a particular design. A better representation would explicitly associate possible detailed expansions with particular plans -- call this lateral expansion. The basic notion is that a plan poses some problems to be solved by filling in further plan details. Some possible solutions to those problems should come to mind with the plan. The latter (lateral) expansions would be good attachment points for commentary indicating how a "plan recognizer" might go about recognizing that a plan might be successfully mapped into a more detailed, laterally-associated plan. For example, the standard top-level plan for a receiver is the simple RF/AF section plan. However, nearly all receivers turn out to be superheterodyne receivers. Consequently, a top-level radio frame would show a laterally (strongly) attached superheterodyne receiver plan (with RF amplifier, converter, IF strip, detector, audio controls, and audio amplifier cascade). Suppose the new, improved super technician wanted to verify that a newly encountered radio was indeed a superheterodyne receiver. He would see commentary attached to the pairing of the RF/AF plan and the superheterodyne plan, indicating that if a mixer and oscillator can be located, there is strong evidence that the receiver at hand fits the superheterodyne paradigm. This reduces the problem to one of recognizing mixers and oscillators -- hopefully a simpler problem.

The rules representing the behavior of plans (when instantiated as plan-fragments) have the usual problem encountered with essentially procedural (imperative) representations -- opacity. How is one to know what the effects of a procedure are other than by executing it and checking to see what happened? The TO-GET and ENTAILment patterns in rule bodies are attempts to give declarative explanations of what the rule code accomplishes. These explanations are just barely "squeaking by" and are inadequate in at least two ways: they generally give

-----  
<sup>1</sup> Recall that in the introductory chapter I argued that the problem of parsing a schematic diagram into a design is basically orthogonal to the problem of reasoning about a design. That remains my belief. There is no reason, however, why the underlying representation should not be helpful in both the reasoning and recognition processes.

minimal descriptions of what the rule code accomplishes, and the rule code itself may affect a number of mutually independent changes in the world model as represented by the state of the data base. If WATSON had been wrought in the predicate calculus paradigm with a single rule of inference, this would not have been a problem since rules would all have been logical implications. The code body of a rule is a new rule of inference which is typically a compilation of a sub-tree of deductions together with a generous portion of context sensing control structure to decide on the applicability of the rule. By using such powerful rules, a great deal of the information -- implicit in the partial order of antecedents and consequents of a (relatively) simple inference rule such as resolution -- is thrown away.

Consider, for example, the narrow-band RF amplifier of figure 2.13. It has two essential functions: to amplify and to squelch signals not within some stated band-width. WATSON's current procedural representation of this amplifier tests the incoming signal for appropriate frequency properties. If they meet specifications, an appropriate output signal is produced in the data base. Otherwise, the output remains quiescent. Thus WATSON cannot posit the lack of amplification without the lack of "narrow-band-ness." Seen from the outside, the RF amplifier's squelching function should be independent of its amplification function, even though its realization in the design may in fact cause them to be interdependent<sup>1</sup>. That is to say, the extrinsic rule description of the amplifier should be:

1. If there is a signal at the amplifier's output, it consists of components, each of which is an amplified copy of a similar component at the input.
2. If there is a signal at the amplifier's output, it meets certain frequency criteria in its carrier components.

Furthermore, the rule should make clear that when a carrier component is pulled through the linear amplifier, its modulation components rides along. A better solution to the rule

-----  
<sup>1</sup> Note that the goal of the amplifier would be stated to any reasonable design program as a conjunction of sub-goals.

representation problem is made difficult by the simultaneous desire to preserve different levels of descriptive language, maintain the independence of consequences, deduce the consequences of particular plans *en masse*, and to associate the consequences with plans at lower levels of detail.

The last of WATSON's limitations that I should like to mention concerns the bug abstractions that he carries out. I feel that bug abstraction is WATSON's major weakness. His abstractions are simultaneously narrow and shallow: narrow in the sense that the generalizations done in the abstraction process are not sufficiently general, and shallow in the sense that the descriptions of the bugs do not represent a deep understanding of the underlying mechanisms. I think that the problem of breadth results partly from the inelasticity of WATSON's plan representations that I discussed above. For example, the AGC bug in the case of C422 should be abstractable to any radio design having an AGC feed-back loop to the IF strip, thus including designs with the other AGC as well<sup>1</sup>. Even better, a genius level technician would abstract the bug to a TRF receiver that had an AGC of the same structure as the IF strip in the GR-78. He would reason that since the IF strip is the RF portion of a TRF receiver, by analogy, the bug should go through. This latter intellectual leap requires WATSON to know that the portion of the GR-78 to the right of the converter section in the plan of figure 2.3 operates as a TRF receiver. A slightly less ambitious example of greater depth of abstraction is provided by the broken oscillator in the case of Q301. WATSON provides a superficial description of the bug mechanism by assertion of the failure to achieve the amplification goal (see section 8.1). This implicitly contains an argument supporting the lack of a stable oscillation. A deeper description of the bug would make the argument explicit by asserting that the lack of amplification leads to both the failure to start oscillation and the failure to stabilize the oscillation, giving a scenario of how the oscillator would behave in the face of deviations from the desired output. The ability to extract such descriptions of bugs from the rule structures for plans awaits a more transparent description of such rules.

-----  
<sup>1</sup> The GR-78 in reality has two AGC circuits, only one of which is active in the control configurations that have been considered.

### 12.5 Directions for further research.

The discussion of the last section suggests many lines of research in improvements and extensions of WATSON. I think that it would be especially fruitful to explore the extensions to the representational machinery that would enable WATSON to discover the design from the bare circuit diagram. I think that a system that could take hints from a "trainer" as guidance to finding the design would probably have subsumed much of the representational machinery needed for doing design.

Another avenue of investigation would involve applying WATSON's techniques to other domains. I have already presented a cursory treatment of such an effort in the micro-world of automobiles. How extensive is the class of deliberate artifacts to which such methods are applicable? I should hope that programs, for example, would be understandable by such techniques. Is there a design language for programs other than that provided by the program itself? This is a difficult problem certainly worth investigating.

### Appendix: Index of Terms

AGENT-156	121	DET-160	123
AGENT-159	122	DET-178	133
AGENT-161	123	DISTORT-MODULANT-184	133
AMP	146	DV/DT-72	82
AMP-34	66	FDBK-98	98
AMP-LIMITER-LAW-44	68	FWDREF-191	138
AMPL-118	103	GOAL-207	155
AMPL-119	103	ID-TH-200	152
AMPLIFY-36	67	io-contour	8
AMPLIFY-66	78	ITEM-201	154
		ITEM-203	154
		ITEM-204	154
		ITEM-212	156
		ITEM-213	156
		ITEM-214	156
		ITEM-215	156
B-107	101	JFET-49	71
B-122	107	KCL-57	72
B-132	112	KCL-87	93
B-136	114		
C-ALARMIST-175	130	LS+	13
C-ALARMIST-196	148	LS1	11
C-ALARMIST-198	149	LS2	11
C-ITEM-205	154	LS3	12
C-ITEM-206	154	LS4	13
CAP-BUG-RULE-73	82	MAXWELL	51
CC-112	103	MC-165	124
CC-113	103	MC-166	124
CC-158	121	MC-168	125
CC-162	123	MC-169	125
CC-180	133	MC-170	125
CF-116	103	MC-182	133
CF-117	103	MF-167	124
CF-163	123	MF-171	125
CF-164	123	MF-172	125
CF-181	133	MF-173	125
CONSIST-216	157	MF-183	133
CTL-4	53	MOD-114	103
CTL-5	54	MOD-115	103
DET-155	120	NODE-52	72
DET-157	121	NODE-94	97



# Appendix

181

ob	51	PURP-209	155
OHM-102	98	PURP-210	155
OHM-59	72		
OHM-70	82	Q301-48	70
		Q406-100	98
PARAM-151	117	Q406-105	99
PARAM-152	117	Q407-84	92
PF-19	63		
PF-20	63	R302-55	72
PF-21	63	R454-103	98
PF-3	53	R81AS-90	93
PF-32	64	REG-10	57
PF-33	66	REG-11	57
PF-38	68	REG-12	58
PF-39	68	REG-120	104
PF-63	75	REG-121	104
PF-65	76	REG-127	109
PF-67	81	REG-128	109
PF-74	85	REG-129	110
PF-75	85	REG-13	58
PF-76	85	REG-130	111
PF-77	85	REG-131	111
PF-78	87	REG-135	113
PF-79	91	REG-14	58
PF-80	91	REG-16	60
PF-81	91	REG-17	60
PF-83	92	REG-26	64
PF-88	93	REG-27	64
PF-95	98	REG-28	64
PORT-1	52	REG-29	64
PORT-109	103	REG-30	64
PORT-141	116	REG-31	64
PORT-143	116	REG-61	74
PORT-145	116	REG-62	74
PORT-147	117	REG-9	57
PORT-149	117	RES-101	98
PORT-190	138	RES-104	99
PORT-197	148	RES-106	99
PORT-199	149	RES-40	68
PORT-2	52	RES-41	68
PORT-22	64	RES-42	68
PORT-24	64	RES-43	68
PORT-35	66	RES-45	68
PORT-64	76	RES-46	68
PORT-82	91	RES-50	71
PURP-208	155	RES-51	71

RES-54	72	SIG-150	117
RES-56	72	SIG-153	120
RES-58	72	SIG-154	120
RES-60	72	SIG-174	126
RES-68	81	SIG-176	132
RES-69	82	SIG-177	132
RES-71	82	SIG-179	132
RES-85	93	SIG-18	61
RES-86	93	SIG-23	64
RES-89	93	SIG-25	64
RES-92	96	SIG-6	56
RES-97	98	SIG-7	56
RES-99	98	SIG-8	56
RF-GAIN	146	sign	
rule	13	SYM-VD-96	
RULE-192	140		
RULE-193	140	TANK-LAW-47	
RULE-194	140		
RULE-202	154	VD-91	9
RULE-211	155	VDBIAS-93	90
RULE-37	68	VIC-53	72
s-185	137		
s-186	137		
s-187	137		
s-188	137		
s-189	137		
s-195	141		
sig-108	101		
SIG-110	103		
SIG-111	103		
sig-123	107		
sig-124	107		
sig-125	107		
sig-126	107		
sig-133	112		
sig-134	112		
sig-137	114		
sig-138	114		
sig-139	114		
sig-140	114		
SIG-142	116		
SIG-144	116		
SIG-146	116		
SIG-148	117		
SIG-15	60		

## Bibliography

- American Radio Relay League. 1973. *The radio amateur's handbook*. 50th ed. Newington, Connecticut.
- Brown, A.L. 1974. *Qualitative knowledge, causal reasoning, and the localization of failures -- a proposal for research*. MIT Artificial Intelligence Laboratory Working Paper no. 61. Cambridge.
- Brown, A.L., and Sussman, G.J. 1974. *Localization of failures in radio circuits: a study in causal and teleological reasoning*. MIT Artificial Intelligence Laboratory Memorandum no. 319. Cambridge.
- Brown, J.S.; Burton, R.R.; and Bell, A.G. 1974. *SOPHIE: a sophisticated instructional environment for teaching electronic troubleshooting*. Bolt, Beranek and Newman Report no. 2970. Cambridge.
- Brown, J.S., and Burton, R.R. 1975. Multiple representations of knowledge for tutorial reasoning. In *Representation and understanding: studies in cognitive science*, eds. D.G. Bobrow and A. Collins, pp. 311-50. New York: Academic Press.
- Chang, C.L., and Lee, R.C.T. 1973. *Symbolic logic and mechanical theorem proving*. New York: Academic Press.
- Clarke, K.K., and Hess, D.T. 1971. *Communication circuits: analysis and design*. Reading, Massachusetts: Addison-Wesley.
- Cruz, J.B., and Van Valkenburg, M.E. 1974. *Signals in linear circuits*. Boston: Houghton Mifflin.
- deKleer, J. Structural knowledge in troubleshooting electronic circuits. In *Steps toward a theoretical foundation for complex, knowledge-based CAI*, ed. J.S. Brown. Bolt, Beranek, and Newman Technical Report, forthcoming. Cambridge.
- Dertouzos, M.L. 1967. CIRCAL: on-line circuit design. *Proceedings of the IEEE*, 55, pp. 637-54.
- Freeman, P., and Newell, A. A model for functional reasoning in design. In *Proceedings of the second international joint conference on artificial intelligence*, pp. 621-33, London: British Computer Society.
- Gabel, R.A., and Roberts, R.A. 1973. *Signals and linear systems*. New York: Wiley.
- Goldstein, I.P. 1974. *Understanding simple picture programs*. MIT Artificial Intelligence Laboratory Technical Report no. 294. Cambridge.

- . 1975. Understanding simple picture programs. *Artificial Intelligence*, 6, pp. 249-88.
- Gray, P.E., and Searle, C.L. 1969. *Electronic principles: physics, models, and circuits*. New York: Wiley.
- Heath Company. 1969. *Model GR-78 general coverage receiver Heathkit assembly manual*. Benton Harbor, Michigan.
- Hewitt, C.E. 1971. *Description and theoretical analysis (using schemata) of PLANNER: a language for proving theorems and manipulating models in a robot*. MIT Artificial Intelligence Technical Report no. 258. Cambridge.
- McCarthy, J. 1968. Programs with common sense. In *Semantic information processing*, ed. M.L. Minsky, pp. 401-18. Cambridge: MIT Press.
- McCarthy, J.; Abrahams, P.W.; Edwards, D.J.; Hart, T.P.; and Levin, M.I. 1965. *LISP 1.5 programmer's manual*. 2nd ed. Cambridge: MIT Press.
- McDermott, D.V. 1974. *Advice on the fast-paced world of electronics*. MIT Artificial Intelligence Laboratory Working Paper no. 71. Cambridge.
- . 1976. *Flexibility and efficiency in a computer program for designing circuits*. Ph.D. dissertation, Massachusetts Institute of Technology. Cambridge.
- McDermott, D.V., and Sussman, G.J. 1974. *The CONNIVER reference manual*. 2nd ed. MIT Artificial Intelligence Laboratory Memorandum no. 259a. Cambridge.
- Minsky, M.L. 1975. A framework for representing knowledge. In *The psychology of computer vision*, ed. P.H. Winston, pp. 211-77. New York: McGraw-Hill.
- Moon, D.A. 1974. *MACLISP reference manual*. MIT Project MAC. Cambridge.
- Moore, R.C. 1975. *Reasoning from incomplete knowledge in a procedural deduction system*. MIT Artificial Intelligence Laboratory Technical Report no. 347. Cambridge.
- Newell, A. 1962. Some problems of basic organization in problem-solving programs. In *Self-organizing systems*, eds. M.C. Yovits, G.T. Jacobi, and G.D. Goldstein, pp. 393-423. Washington, D.C.: Spartan Books.
- Newell, A., and Simon, H. A. 1976. Computer science as empirical inquiry: symbols and search. *Communications of the Association for Computing Machinery*, 19, pp. 113-26.
- Penfield, P. 1971. *MARTHA user's manual*. Cambridge, MIT Press.
- Roylance, G.L. 1975. *Anthropomorphic circuit analysis*. Bachelor's thesis, Massachusetts Institute of Technology. Cambridge.
- Rubin, A.D. 1975. *Hypothesis formation and evaluation in medical diagnosis*. MIT Artificial Intelligence Laboratory Technical Report no. 316. Cambridge.
- Sacerdoti, E.D. 1975a. The nonlinear nature of plans. In *Proceedings of the fourth international joint conference on artificial intelligence*. pp. 206-13, International Council on Artificial Intelligence.

- . 1975b. *A structure for plans and behavior*. SRI Artificial Intelligence Center Technical Note no. 103. Menlo Park, California.
- Senturia, S.D., and Wedlock, B.D. 1975. *Electronic circuits and applications*. New York: Wiley.
- Shortliffe, E.H. 1974. *MYCIN: a rule-based computer program for advising physicians regarding antimicrobial therapy selection*. Stanford Artificial Intelligence Laboratory Memorandum no. 251. Stanford, California.
- . 1976. *Computer-based medical consultations: MYCIN*. New York: American Elsevier.
- Sussman, G.J. 1975. *A computer model of skill acquisition*. New York: American Elsevier.
- Sussman, G.J., and Stallman, R.M. 1975. Heuristic techniques in computer aided circuit analysis. *IEEE Transactions on Circuits and Systems*, CAS-22, pp. 857-65.
- Watson, J. 1970. *Semiconductor circuit design*. 2nd ed. London: Adam Hilger.
- Winston, P.H. 1975. Learning structural descriptions from examples. In *The psychology of computer vision*, ed. P.H. Winston, pp. 157-209. New York: McGraw-Hill. end

*This blank page was inserted to preserve pagination.*

**CS-TR Scanning Project**  
**Document Control Form**

Date : 4 / 4 / 96

Report # AI-TR-362

Each of the following should be identified by a checkmark:

Originating Department:

- ☒ Artificial Intelligence Laboratory (AI)  
☐ Laboratory for Computer Science (LCS)

Document Type:

- ☒ Technical Report (TR) ☐ Technical Memo (TM)  
☐ Other: \_\_\_\_\_

**Document Information**

Number of pages: 195 (22 IMAGES)

Not to include DOD forms, printer instructions, etc... original pages only.

Originals are:

☒ Single-sided or

☐ Double-sided

Intended to be printed as :

☐ Single-sided or

☒ Double-sided

Print type:

- ☐ Typewriter ☐ Offset Press ☐ Laser Print  
☐ InkJet Printer ☐ Unknown ☒ Other: \_\_\_\_\_

Check each if included with document:

- ☒ DOD Form ☒ Funding Agent Form ☒ Cover Page  
☐ Spine ☐ Printers Notes ☐ Photo negatives  
☐ Other: \_\_\_\_\_

Page Data:

Blank Pages (by page number): FOLLOW TITLE PAGE, QUOTE PAGE, IX.

Photographs/Tonal Material (by page number): \_\_\_\_\_

Other (note description/page number):

Description :

Page Number:

① IMAGE MAP: (1-195) UN# KD TITLE PAGE, UN# BLANK, UN# ABSTRACT,  
UN# ED ACKN, UN# QUOTE PAGE, UN# BLANK,  
Vii - IX, UN# BLANK, 1-185

(196-202) SCAN CONTROL, COVER, FUNDING AGENT, DOD, TRGT'S (3)  
② PAGES NUMBERED 1, 2, 19 AND 20 ARE XEROX REPLACEMENT PAGES.

Scanning Agent Signoff:

Date Received: 4/4/96 Date Scanned: 4/16/96

Date Returned: 4/18/96

Scanning Agent Signature: Michael W. Cobb

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AI-TR-362	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle)  Qualitative Knowledge, Causal Reasoning, and the Localization of Failures		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Allen L. Brown, Jr.		8. CONTRACT OR GRANT NUMBER(s) N00014-70-A-0362-0003
9. PERFORMING ORGANIZATION NAME AND ADDRESS Artificial Intelligence Laboratory 545 Technology Square Cambridge, Massachusetts 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Blvd Arlington, Virginia 22209		12. REPORT DATE November 1976
		13. NUMBER OF PAGES 198
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research Information Systems Arlington, Virginia 22217		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		16. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Distribution of this document is unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES  None		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  Automatic Maintenance                      Artificial Intelligence Electrical Analysis                          Computer-edited Design Signal Tracing		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  This report investigates some techniques appropriate to representing the knowledge necessary for understanding a class of electronic machines -- radio receivers. A computational performance model "WATSON" is presented. WATSON's task is to isolate failures in radio receivers whose principles of operation have been appropriately described in the knowledge base.		



# Scanning Agent Identification Target

Scanning of this document was supported in part by the **Corporation for National Research Initiatives**, using funds from the **Advanced Research Projects Agency of the United States Government** under Grant: **MDA972-92-J1029**.

The scanning agent for this project was the **Document Services** department of the **M.I.T Libraries**. Technical support for this project was also provided by the **M.I.T. Laboratory for Computer Sciences**.

